Modeling and Simulation of Axisymmetric Stagnation Flames

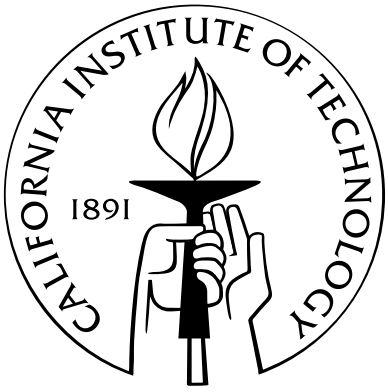
Thesis by

Kazuo Sone

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy



California Institute of Technology

Pasadena, California

2007

(Defended May 18, 2007)

To Juri

Acknowledgements

I would like to thank my advisors, Professor Paul Dimotakis and Professor Daniel Meiron, for guiding me throughout my doctoral research. It was truly a privilege to be able to work with them. This research project was very challenging, yet I was able to complete it because of their guidance. I was constantly inspired by their knowledge and intelligence. I would also like to thank Professors David Goodwin, Joseph Shepherd, and Dale Pullin for serving on my thesis committee and providing many suggestions that improved the quality of this thesis. Outside of the thesis committee, I am indebted to Professors Anthony Leonard and Tim Colonius for the discussion of the polar axis treatment that was useful in developing the new basis functions used in this study.

This research project is running in parallel with experiments. In addition to providing experimental data, I had many discussions on the subject with Jeff Bergthorson, and that significantly influenced this thesis. Laurent Benezech also helped me with additional data, some post-processing of data and by providing some of one-dimensional solutions used in this study. Many members of the Compressible Turbulence group gave me input including Carlos Pantano, David Hill, George Matheou, Trent Mattner, and Patric Hung. I am grateful for the discussions with them.

The members of the Center for Advanced Computing Research at Caltech and Livermore computing were very supportive by maintaining and updating computational resources, as well as giving me a hand in computing. In particular, I am very thankful to Michael Aivazis, Sharon Brunett, Mark Bartelt, Julian Cummings, and Jan Lindheim. Computer support by Dan Lang and administrative support by Christina Mojahedi at Professor Dimotakis' group are also appreciated. This project was funded by the Caltech Advanced Simulation and Computing (ASC) program of the Center for Simulation of Dynamic Response of Materials, and the generous computing resources are also provided by the program, with additional computing resources provided by the Air Force Office of Scientific Research.

Finally, I would like to thank my wife, Juri for supporting and encouraging me throughout my doctoral studentship.

Abstract

Laminar flame modeling is an important element in turbulent combustion research. The accuracy of a turbulent combustion model is highly dependent upon our understanding of the laminar flames and their behavior in many situations. How much we understand various phenomena can only be measured by a model that describes the phenomena and by how well the model describes and predicts them. One of the most commonly used methane combustion models is GRI-Mech 3.0. However, how well the model describes the reacting flow phenomena is still uncertain, even after many attempts to validate the model or quantify uncertainties. This is because, in flames, chemisty is coupled with fluid mechanics, thermodynamics, and transport process, and the separation of one from another is not easy, if at all possible.

In the present study, the behavior of laminar flames under different aerodynamic and thermodynamic conditions is studied numerically in a stagnation-flow configuration. The present study follows an experimental study by J. Bergthorson conducted earlier in our group. In numerical study of reacting flows, a one-dimensional model is commonly used to assess the performances of chemical kinetics models. The model describes stagnation flames along the symmetric axis through several key assumptions. One such assumption is a uniform pressure-eigenvalue assumption, *i.e.*, that the curvature of the pressure field is uniform throughout. Although it is shown that this assumption does not hold through more sophisticated numerical studies capable of a two-dimensional description, it is shown that the model works reasonably well in the case of non-reacting (cold) flow and diluted hydrogen flames. However, how well the assumption holds and whether or not the model approximates hydrocarbon flames well are not known. The present study employs the full chemical kinetics model of methane combustion, and a realistic transport model that accommodates differential-diffusion effects within an axisymmetric two-dimensional flow modeling. This allows direct comparisons of two-dimensional and one-dimensional models,

as well as numerical and experimental data, to quantify modeling errors that arise from the use of one-dimensional hydrodynamics model and the chemical-kinetics model.

In order to make such a numerical study possible, the spectral element method is reformulated to accommodate the large density variations in methane reacting flows. In addition, a new axisymmetric basis function set for the spectral element method that satisfies the correct behavior near the axis is developed that avoids the well-known singular behavior there. This basis function satisfies all the parity requirements, by construction, that axisymmetric fields must meet. To accomodate computationally expensive detailed methane combustion and transport models, efficient integration techniques are developed to accurately model axisymmetric reacting flow within a reasonable amount of computational time. The numerical method is implemented using an object-oriented programming technique, and the resulting computer program is verified with several different methods.

First, cold-flow simulation is conducted to understand the nature of the underlying flow field without chemical reactions. It is shown that detailed modeling of the experimental apparatus is important for a direct comparison of numerical simulation and experiments to be meaningful.

Reacting flow simulations are conducted in three phases: one-dimensional simulations by Cantera, two-dimensional simulations with an idealized representation of the experimental configuration, and finally, simulations with full details of experimental setup. It is shown that, although the plug-flow boundary condition cannot be used, as is, to predict flame locations, the model can reliably be used to predict flame speed under strain. Through a direct simulation of laboratory flames that allows direct comparison to experimental data, the present study then shows variances with the commonly used GRI-Mech 3.0 chemical kinetics model. It is shown that the methane combustion model based on GRI-Mech 3.0 works well for methane-air mixtures near stoichiometry. However, GRI-Mech 3.0 leads to an overprediction of laminar flame speed for lean mixtures and an underprediction for rich mixtures. This result is slightly different from conclusions drawn in previous work, in which experimental data are compared with a one-dimensional numerical solution. Detailed analysis reveals that flame speed is sensitive to even slight flame front curvature as well as to its finite extension in the radial direction. Neither of these can be incorporated in one-dimensional flow modeling.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Often, it is people rather than the technology itself who pose the greater challenge to solving an otherwise technical problem.—John Lakos

1.1 Motivation

The behavior of laminar flames and insights into such phenomena have many implications to our understanding of turbulent as well as laminar flames. Even when the underlying flow is turbulent, as is the case in internal combustion or jet engines, the flame in the case of fast kinetics (high Damköhler number) is considered as an ensemble of stretched laminar flames called flamelets (Williams, 1975). Recent advances in flamelet modeling are due, in part, to the understanding of the laminar flame structure and behaviors both experimentally and numerically (Law, 1988; Law & Sung, 2000; Williams, 2000). In particular, the laminar flame speed under the influence of aerodynamic effects such as stretch and strain, flame curvature, or heat losses has been studied experimentally. Numerical simulations have also previously been employed to model such flame behavior, but with limited success. This is mostly because simulation of combustion phenomena and its interaction with fluid mechanics is computationally expensive and oversimplified models required to obtain numerical results within a practical amount of time were insufficient to reveal important aspects of the full phenomenology.

Simulation is used everywhere. Automobile and aerospace industries are among the major users of Computer-Aided Engineering (CAE), including Computational Fluid Dynamics (CFD), but the application of CAE goes beyond such traditional users and now extends to semiconductors, pharmaceuticals, food processing (Lange, 2007), high-performance sport-

ing goods (McKee, 2004), and so on. The use of computer simulations in science and engineering is now widespread. For example, simulations used in product design allows engineers to develop products quickly for market. The short development cycle can lead to low R&D costs and therefore cost competitiveness in the market, in addition to the direct savings from replacing some expensive laboratory experiments. Sometimes, even when cost does not come at the top of the priority list, simulations are the only possibility to conduct science, due to safety concerns when highly toxic or explosive materials are involved, or due to conditions which are either impossible or difficult to attain in the laboratory.

However, there is a down side to this rapid growth of the application of computational engineering. Toyota Motor Corp., which is known for its quality automobiles, recently suffered multiple waves of recalls, and after internal investigation, the president of the company had to say (Shirouzu, 2006):

> "We relied on computer-aided engineering and other computer analysis and didn't conduct as many quality checks as we should have."

It reminds us that simulation software—when used naively or incorrectly—can lead to a devastating result. A main part of the problem lies in the growing complications in the software.

Often, simulations are used for complicated problems in which verification of the correctness of the numerical results is not easy, and this makes simulation software prone to human mistakes, including programming errors (commonly referred to as 'bugs') and misuse. Even though it is impossible to prove the correctness of software[†] it is still possible to develop useful and reliable software. For example, there is a formal protocol called Software Quality Assurance (SQA) to ensure the reliability of software, as well as techniques to develop large-scale software such as Object-Oriented Programming (OOP) and design patterns (Gamma *et al.*, 1994). Every computational scientific work needs to address this issue, and the approach used in this study is described in Chapter 3.

Besides such human errors, there are two additional kinds of errors in every computer simulation: modeling errors and numerical errors (including discretization errors, convergence errors, and round-off errors). Modeling errors are a deviation of the mathematical

[†] *"Complete testing, erroneously used to mean 100% branch coverage. The notion is specific to a test selection criterion: i.e., testing is 'complete' when the tests specified by the criterion have been passed. Absolutely complete testing is impossible."*—from "Software Testing Techniques" by Beizer (1990).

description of the physical system (*i.e.*, model) from the real physical system. Some flow systems offer a better description than others. For example, a uniform-density, low Mach number flow is believed to be well described by the incompressible, uniform-density Navier-Stokes equations, and simulations employing such validated mathematical models are at least as accurate as thoughtfully and carefully conducted experiments. Indeed, some authors have attempted to estimate and quantify measurement errors from numerical data. However, for some types of flow systems, we do not yet have such a sufficiently high-fidelity mathematical description. One such example is chemically reacting flow systems, which is the topic of the present study. Although significant work has been done in this area, there has been only limited progress in obtaining reliable chemical kinetics models for the combustion of hydrocarbon fuels. There have been much theoretical, experimental, and numerical work on laminar flames and flamelets. However, theoretical work has been limited to cases with oversimplified assumptions, such as flames with no heat release, or infinitely thin flames. The numerical work has relied on reduced chemistry models or oversimplified hydrodynamic models.

Only few recent reports exist that used a detailed hydrocarbon combustion model with realistic fluid mechanical models. For example, Najm & Knio (2005) proposed an operator-splitting scheme for low Mach number, chemically reacting flows with GRI-Mech 1.2 that involves 32 species and 177 reactions[‡] to demonstrate their algorithm. Bell *et al.* (2005a) used the reaction mechanism of Glarborg *et al.* (2000) that includes 65 species and 447 elementary reactions to simulate a laminar diffusion flame. Many other simulations of combustion phenomena using multi-dimensional models have been reported using reduced chemistry models. For a simulation of a turbulent flame, Bell *et al.* (2005b) used a subset of GRI-Mech 1.2 mechanism that includes 20 species and 84 elementary reactions to simulate a turbulent V-flame. This is probably one of the biggest simulation in terms of the number of species and reactions used in turbulent combustion simulations, and while the resutls are very impressive, the numerically predicted angle of the V-flame is wider than that observed in the experiment by 10%. There are many chemical kinetics models proposed for methane combustion, but the lack of an appropriate framework—a multi-dimensional simulation of laboratory-scale flames that allows direct comparison of numerical data to experimental ones—limits our understanding of laminar flames, and consequently turbulent flames.

[‡]These numbers include argon whereas numbers in Table 1.1 do not.

Table 1.1: The chemical kinetics models. M is the total number of species involved, and K stands for the number of reactions in the mechanism. The reaction model number indicates reactants considered, *e.g.*, HCN indicates Hydrogen, Carbon (Hydrocarbons) and Nitrogen oxidation reaction set. Reduced mechanisms are indicated by lower-case model IDs. This list is by no means complete.

Model ID.	Name	M	K	References
HCN1	GRI-Mech 3.0	53	325	(Smith *et al.*)
HCN2	GRI-Mech 2.11	49	277	Predecessor of HCN1
HCN3	Glarborg00	65	447	(Glarborg *et al.*, 2000; Bell *et al.*, 2005a)
HC1	GRI-Mech 3.0	35	217	Sub-mechanism of HCN1
HC2	GRI-Mech 1.2	31	175	Predecessor of HCN2
HC3	SD05 (Rel. 03/10)	39	175	(San Diego mechanism)
HC4	DLW99	71	469	Based on HC2, (Davis *et al.*, 1999, 2002b)
HC5	Wang99	52	367	(Wang *et al.*, 1999)
HC6	Marinov99	57	383	(Marinov, 1999)
HC7	WF97	33	192	(Wang & Frenklach, 1997)
HC8	Tan94	78	473	(Tan *et al.*, 1994)
HC9	EDL92	30	171	(Egolfopoulos *et al.*, 1992)
hc1	Smooke92	26	83	(Smooke *et al.*, 1992; Day & Bell, 2000)
hc2	Smooke86	16	46	(Smooke *et al.*, 1986)
hc3	DRM-19	20	84	Subset of HC2, Bell *et al.* (2005b)
hc4	1-step model	4	1	(Westbrook & Dryer, 1981)
H1	GRI-Mech 3.0	9	28	Sub-mechanism of HC1
H2	SD05 (Rel. 03/10)	9	22	Sub-mechanism of HC3
H3	DLW99	9	28	Sub-mechanism of HC4
H4	YDR91	9	19	(Yetter *et al.*, 1991; Frouzakis *et al.*, 1998)
H5	MW88	9	37	(Maas & Warnatz, 1988)

This is very unfortunate because our everyday life relies heavily on combustion: furnance, gas turbine, and automobile engines, just to name a few. Presently in this country, slightly more than three quarters of our electricity supply relies on combustion[§].

There are several chemistry models for methane combustion available. For example, Egolfopoulos & Dimotakis (2001) used and compared several natural gas combustion models including those of Tan *et al.* (1994), GRI-Mech (Smith *et al.*), Wang & Frenklach (1997), Marinov (1999) and Wang *et al.* (1999). Bergthorson (2005) used San Diego mechanism, and a C-3 mechanism by Davis *et al.* (1999), in addition to GRI-Mech 3.0, to study laminar flame speed at various conditions. These mechanisms are summarized in Table 1.1. All are comprised of a seemingly sufficient set of elementary reactions. However, numerical

[§]Energy Information Administration, Form EIA-906, "Power Plant Report."

results obtained from each of these models vary significantly (Egolfopoulos & Dimotakis, 2001; Bergthorson, 2005). Many of them take elementary reaction parameters from experimental data and evaluate rate coefficients of a single reaction to assemble a complete set of elementary reactions. However, this approach can fail due to correlations in uncertainties in the parameters, and the best-fit values to the individual parameters do not necessarily comprises the best chemical kinetics model (Frenklach *et al.*, 1992). GRI-Mech takes a slightly different approach that is based on a systematic optimization method called solution mapping (Frenklach *et al.*, 1992), through which the mechanism is optimized to several independent flame conditions, including shock-tube ignition delay, methyl radical concentration, and laminar premixed flame speeds. As a consequence, this mechanism is regarded as the best methane combustion model without problem-specific fine tuning of parameters.

These chemical kinetics models have been primarily used in numerical studies that employ a one-dimensional formulation developed by Kee *et al.* (1988) that represents a significant advance in combustion research. Inclusion of these detailed methane combustion models in a multi-dimensional numerical simulation is computationally expensive, and numerical studies of laminar flames in multi-dimensions have typically used reduced mechanisms such as (Smooke *et al.*, 1986, 1992) that include a subset of all species involved and reduced reaction sets. It is not obvious if comparison of one-dimensional numerical solutions to laboratory-generated experimental data is valid, even when they agree. When they do not agree, it is not clear if the chemical kinetics model is the cause of error or the simplified fluid mechanics model in the one-dimensional model is responsible, or some combination of the two.

There are three problems here. One is that there has been no appropriate validation of chemical kinetics models that gives us confidence in these models and in the numerical results in studying flame behavior, or designing reaction systems. This is primarily because of the difficulty in realistic multidimensional simulations of hydrocarbon flames because of excessive computational costs. An algorithm and software that allow direct simulation of chemically reacting flows are required, and this is the second problem. The third problem is the reliability of the simulation software itself. Knupp & Salari (2003) collected appropriate procedures for testing programs for computational science and engineering, but attention has been paid to this subject only very recently.

1.2 Objective

The primary objective of the present study is to understand the validity and applicability of the GRI-Mech 3.0 chemistry model and to make progress toward a universal natural-gas (methane) combustion model that works for every fuel in every situation, including both rich and lean conditions, as well as under low or high pressure.

To achieve this goal, a computational framework that allows us to evaluate the accuracy of the chemistry model is required. An efficient and accurate algorithm to simulate a laboratory flame directly, including appropriate initial and boundary conditions, was developed. The spectral element method originally developed by Patera (1984) for incompressible flow will be extended to accommodate capabilities to simulate chemically reacting flows with large density variation efficiently. Second, error sources to computational simulations will be identified, and then the list of methods to quantify or identify errors stemming from each error source will be discussed. It will be subsequently shown that the code used in this study is correct and accurate with respect to the selected test criteria. Third, the developed computing framework will be used to study the behavior of a chemical-kinetics model. Namely the behavior of GRI-Mech at various strain and equivalence ratios will be studied for premixed methane flames. The aim is to clarify the validity of one-dimensional modeling of stagnation flow so that computationally simpler, one-dimensional models can be used judiciously. Through these developments, how aerodynamic effects such as strain, stretch, dilatation, and flow non-uniformity that creates flame front curvature can affect flame speed and flame behavior are investigated. These studies are of interest in their own light, but can also provides a good validation of the GRI-Mech 3.0 model, if the numerical results are in agreement with a theory or experimental prediction.

Chapter 2

The simulation methods

Simulation: *3. The technique of imitating the behaviour of some situation or process (whether economic, military, mechanical, etc.) by means of a suitably analogous situation or apparatus, esp. for the purpose of study or personnel training. Freq. attrib.* —Oxford English Dictionary (emphasis added by the author)

2.1 Introduction

As the definition of the word "simulation" suggests, for a numerical study to also be a simulation it must be a "suitably analogous" situation or apparatus. Many numerical studies do not include every detail of the situation or apparatus, or do not "imitate the behavior" at a reasonable accuracy (as inclusion of curved geometry, for example, usually deteriorates the accuracy of numerical methods). For example, Bell *et al.* (2005b) computed a laboratory-scale turbulent V-flame using a reduced methane chemistry model. Although this calculation used a large number of reaction and species sets compared to other turbulent combustion simulations, the numerically predicted angle of V-flame was wider than experiment by 10%. Use of reduced chemistry is certainly one cause of error, but more importantly, this might have been caused by not modeling the nozzle exit area where the flow around the anchoring rod is modified by the attached V-flame, as noted by the authors. On the other hand, the KIVA code (Amsden *et al.*, 1985, 1989; Amsden, 1993, 1997, 1999) developed at the Los Alamos National Laboratory over decades is a significant achievement in the simulation of mixing and combustion of internal combustion engines and used by many researchers including Han & Reitz (1995), Celik *et al.* (2000), and Sone & Menon (2003). Although the code has the capability to accomodate arbitrary-shaped cylinders as

well as valve and piston movements, it is very diffusive due to its first-order temporal and spatial accuracy, and its reliability is uncertain.

As the title of this thesis suggests, the present study *is* about a simulation of specific laboratory phenomena, and techniques were developed with this purpose in mind to model flames in a laboratory with a suitably analogous setup with sufficient accuracy. The detailed modeling of the experimental setup have not been done in previous works on numerical studies of laminar flames. The methodology used in this study is described in this chapter. First, the governing equations of chemically reacting flow are reviewed. Then the numerical method employed in this study, including discretization of the governing equations, is introduced. Details on the implementation of the algorithm will be deferred to the next chapter.

2.2 Mathematical models of reacting flows

2.2.1 The governing equations for axisymmetric flow at low Mach number

The governing equations of the fluid mechanics of chemically reacting flow are the compressible form of the Navier-Stokes equations. The unsteady equations have been used in many studies (Amsden *et al.*, 1989; Kim *et al.*, 1999; Haworth & Jansen, 2000). However, these studies are mostly concerned with turbulent reacting flow, and the trouble is that the laminar methane flames exhibit flame speeds far less than those of turbulent flames. Therefore, the flow time scale is usually far smaller than that of acoustic waves, and the explicit integration of equations for compressible flows leads to a very restrictive time step size compared to the time scale of the phenomena. In addition, if the compressible equations are used for low-speed flows, the pressure-gradient term becomes singular as the Mach number approaches to zero and some sort of corrections must be applied (Ramshaw *et al.*, 1985; Amsden *et al.*, 1989). To circumvent these issues, self-consistent equations for low-speed reacting flow have been derived (Majda & Sethian, 1985; McMurtry *et al.*, 1986) using low Mach number asymptotic analysis, and similar equations are derived and used by many others to study laminar flames. This formulation, often called a low Mach number formulation, or a zero Mach number formulation, effectively removes the propagation of sound waves as a means of equilibrating pressure, and thus allows larger timestep size, replacing this dynamical step by a Poisson-solver for pressure. Suppose for example, for the current

problem, the flame thickness is 1.0×10^{-4} m and 10 collocation points are necessary to resolve the flame. The advection timescale is $\Delta t^{\text{Adv}}_{\text{compress}} \sim 3.0 \times 10^{-8}$ sec. However, when a low Mach number formulation is employed, the speed of sound constraint disappears from the denominator of the advection timescale and $\Delta t^{\text{Adv}}_{\text{lowMach}} \sim 1.0 \times 10^{-5}$ sec. Significant savings in computational time can be achieved when the interaction of acoustic waves and flame is not of interest or importance.

The low Mach number formulation used in this study is conceptually different in how bulk viscosity is treated. The derivation of the following equations for density, velocity, temperature, and species used in the present study is described in Appendix A.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{2.1a}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p^* + \mathbf{L}^e(\mathbf{u}) + \mathbf{L}^i(\mathbf{u}) + \mathbf{f}, \tag{2.1b}$$

$$\frac{\partial Y_{\mathfrak{m}}}{\partial t} + \mathbf{u} \cdot \nabla Y_{\mathfrak{m}} = D_{\mathfrak{m}} \nabla^2 Y_{\mathfrak{m}} + \frac{1}{\rho} \nabla \rho D_{\mathfrak{m}} \cdot \nabla Y_{\mathfrak{m}} + \dot{\omega}_{\mathfrak{m}}, \tag{2.1c}$$

$$\rho C_{p,\text{mix}} \frac{DT}{Dt} = \lambda_T \nabla^2 T + \nabla \lambda_T \cdot \nabla T - \rho \sum_m h_{\mathfrak{m}} \dot{\omega}_{\mathfrak{m}}, \text{ and} \tag{2.1d}$$

$$\rho = p_0 / (\bar{R}_{\text{gas}} T), \tag{2.1e}$$

where ρ is the density, $\mathbf{u}$ is the velocity, T is the temperature, and $Y_{\mathfrak{m}}$ is the mass fraction of species $\mathfrak{m}$.

The viscous terms in the momentum equation are

$$\mathbf{L}^i(\mathbf{u}) = \frac{\mu}{\rho} \nabla^2 \mathbf{u} \tag{2.2a}$$

$$\mathbf{L}^e(\mathbf{u}) = \frac{\mu}{\rho} \nabla \left[\nabla \cdot \mathbf{u} \right] + \frac{\nabla \mu}{\rho} \cdot \left[\nabla \mathbf{u} + \nabla \mathbf{u}^t \right]. \tag{2.2b}$$

These equations exhibit a dependence on the external force, $\mathbf{f}$, the species-production rate, $\dot{\omega}_{\mathfrak{m}}$, the enthalpy of the species $\mathfrak{m}$, $h_{\mathfrak{m}}$, mixture-averaged specific heat capacity, $C_{p,\text{mix}}$, ambient pressure, p_0, and the specific gas constant, $\bar{R}_{\text{gas}}$—as well as on transport properties, μ and λ_T, which are the viscosity and thermal conductivity of the gas mixture, respectively, and $D_{\mathfrak{m}}$ the diffusion coefficient of species $\mathfrak{m}$.

In the momentum equation, p^* is the perturbational pressure, which is different from p_0

in the equation of state, and defined by

$$p^*(x,t) = (p(x,t) - p_0) - \left(\mu_{\mathrm{B}} - \frac{2}{3}\mu\right) \nabla \cdot \mathbf{u} \tag{2.3}$$

where p_0 is the leading-order pressure term, which is an ambient pressure and treated as a constant (101.3 kPa) throughout this study, and μ_{B} is the bulk viscosity. Note that p^* includes bulk-viscosity effects. It is possible to factor this term out, which leads to a different value of p^*, but one still obtains the same velocity field. This is because inclusion of the last term of Eqn.(2.3) in the momentum equation (2.1b), rather than in Eqn.(2.3), corrects p^* such that it will cancel the term when the momentum equations are integrated. When this term is comparable to the ambient pressure, p_0, the governing equation itself is invalid and the compressible form of the equations must be used. In the present study, *a posteriori* analysis shows the divergence is on the order of 10^3/s within the flame, and even when μ_{B}/μ is on the order of 10^3, this term is still on the order of 10, which is far less than the ambient pressure, and therefore, it seems this is a reasonable formulation for the current problem.

The species and temperature transport equations used here assume the Fickian diffusion model with mixture-averaged transport properties. No Soret, Dufour, or pressure-gradient diffusion effects are considered. Implications of these assumptions are discussed in Appendix A.

These equations seemingly overspecify the density field. There is an evolution equation, Eqn.(2.1a), also the equation of state, Eqn.(A.28e). On the other hand, there is no equation for the pressure (p^*). To resolve this imbalance of variables and equations, the pressure Poisson equation (PPE) is derived and employed. The derivation of PPE depends upon the discretization of the momentum equations and will be discussed later in the chapter.

In the present study, these unsteady equations are integrated until the numerical solution reaches steady state.

In addition to the governing equations for reacting flows at low Mach numbers, two additional models are used in this study. One is a model for an axisymmetric uniform density flow to study non-reacting flows, and the other is a one-dimensional model of stagnation flames developed by Kee *et al.* (1988).

2.2.2 Axisymmetric incompressible uniform-density flow

With the additional assumption of uniform density, the following equations can be derived from the above low Mach number equations. The governing equations for uniform-density flow are the Navier-Stokes equations with a divergence-free constraint on the velocity field, and those are the equations for which the spectral element method was originally developed by Patera (1984). These equations are later used to analyze cold (non-reacting) flows to evaluate discrepancies between simulations and experiments when no chemical reactions are involved:

$$\nabla \cdot \mathbf{u} = 0, \tag{2.4a}$$

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{N}(\mathbf{u}) - \nabla \tilde{p} + \nu \mathbf{L}(\mathbf{u}). \tag{2.4b}$$

Here, $\mathbf{L}$ is a linear operator ∇^2, and $\mathbf{N}$ is a non-linear operator whose form is given later. In the momentum equation, $\tilde{p}$ denotes the scaled perturbational pressure, p^*, divided by density, and is frequently referred to as the kinematic pressure.

One of the difficulties in solving these equations numerically is the divergence-free constraint, as discussed in Gresho (1991). In the spectral element method, this constraint is satisfied through pressure projection along with a splitting method (Ianenko, 1971). Karniadakis *et al.* (1991) derived a high-order splitting method that preserves high-order temporal accuracy. However, more importantly, their splitting method is consistent with the divergence-free constraint and is used in this study as well.

2.2.3 Axisymmetric one-dimensional model for reacting flow

In an important contribution, Kee *et al.* (1988) proposed a further simplification to axisymmetric stagnation-flow modeling. In addition to the symmetry, they assumed a particular form of the streamfunction, $\Psi(x, r) = r^2 U(x)$, which is the leading order of the Taylor series of an arbitrary axisymmetric streamfunction. The third assumption is a constant pressure eigenvalue, *i.e.*, $1/r(\partial p/\partial r)$ is assumed constant throughout the domain. The first two assumptions effectively separate the variables in the definition of the velocity field,

$$u(x) = 2U(x)/\rho(x), \tag{2.5}$$

and

$$v(x,r) = -rU'(x)/\rho(x). \tag{2.6}$$

The last assumption makes it possible to decouple the axial momentum equations and the velocity is determined by the radial momentum balance only,

$$\Lambda \equiv \frac{1}{r}\frac{\partial p}{\partial r} = \frac{d}{dx}\left(\frac{2U}{\rho}\frac{dU}{dx}\right) - \frac{3}{\rho}\left(\frac{dU}{dx}\right)^2 - \frac{d}{dx}\left[\mu\frac{d}{dx}\left(\frac{1}{\rho}\frac{dU}{dx}\right)\right]. \tag{2.7}$$

This equation can be obtained by dividing the radial momentum equation, Eqn.(A.47), by r, and using the Stokes hypothesis ($\mu_{\mathrm{B}} = 0$). Let V be a spreading rate, dv/dr, which is related to the axial velocity gradient through the continuity equation. The above equation becomes

$$V = -\frac{1}{2\rho}\frac{d(\rho u)}{dx}, \tag{2.8a}$$

$$\Lambda = -\frac{d}{dx}(\rho u V) - 3\rho V^2 + \frac{d}{dx}\left(\mu\frac{dV}{dx}\right). \tag{2.8b}$$

These equations, along with four boundary conditions, typically u and V specified at both ends, yield a one-dimensional solution of the stagnation flame. The axial momentum equation may be used to recover the pressure field from the velocity field if desired:

$$\frac{\partial p}{\partial x} = -4U\frac{d}{dx}\left(\frac{U}{\rho}\right) - 2\mu\frac{d}{dx}\left(\frac{1}{\rho}\frac{dU}{dx}\right) + \frac{4}{3}\frac{d}{dx}\left[2\mu\frac{d}{dx}\left(\frac{U}{\rho}\right) + \frac{\mu}{\rho}\frac{dU}{dx}\right]. \tag{2.9}$$

This equation, along with the temperature and the species balance equations along the axis ($\partial/\partial r = 0$ due to axisymmetry), comprise the one-dimensional model. This formulation is relatively simple, and there are ready-to-use codes available such as CHEMKIN or more recently, Cantera (Goodwin, 2003) (both of which implement this model). It has been widely used in the combustion community partly because there have been no practical alternatives in studying combustion problems numerically, since combustion simulations are computer intensive, and inclusion of detailed chemical-kinetics models has been prohibitive in multidimensional simulations in the past.

Although widely used, there is a difficulty with this model. Quoting from the original paper right after Eqn.(6),

$$\frac{\partial}{\partial x}\left(\frac{1}{r}\frac{\partial p}{\partial r}\right) = \frac{1}{r}\frac{\partial}{\partial r}\left(\frac{\partial p}{\partial x}\right) = 0.$$

This is the logic behind assuming that the pressure eigenvalue, $\Lambda = 1/r(\partial p/\partial r)$, is constant. However, the last equality does not hold in general. Since

$$\frac{\partial}{\partial r}\left(\frac{\partial p}{\partial x}\right) \to 0 \tag{2.10}$$

as $r \to 0$, the entire fraction may be finite. When the streamfunction is given by the assumed form, $\partial/\partial x$ is only a function of x as given in Eqn. (2.9), and indeed Λ is a constant. Therefore, how well the one-dimensional model works depends on how well the assumed form of streamfunction is satisfied in a real flow. The consequence of this simplification will be discussed later.

In the present study, the Cantera software package (Goodwin, 2003) is used to solve the above one-dimensional model.

2.2.4 The chemical reaction model

In addition to flow modeling, chemistry source terms and the transport properties that appear in the governing equation must be modeled and evaluated. This part follows practice used in CHEMKIN / Cantera, with more details in Kee *et al.* (2003).

The source term, $\dot{\omega}_{\mathfrak{m}}$, in the species transport equations, Eqn.(2.1c), represents the creation and destruction of a particular species during the combustion process. Suppose there are K reactions and M species. The k-th reaction can be described by

$$\sum_{\mathfrak{m}\in\mathcal{M}} \nu'_{\mathfrak{m}k}\mathfrak{m} \Leftrightarrow \sum_{\mathfrak{m}\in\mathcal{M}} \nu''_{\mathfrak{m}k}\mathfrak{m} \tag{2.11}$$

where $\nu_{\mathfrak{m}k}$ denotes stoichiometric coefficients of species $\mathfrak{m}$ in reaction k, and the set $\mathcal{M}$ contains all species relating to the given reaction system ($M = |\mathcal{M}|$). The superscript (r) denotes reactants and (p) denotes products.

Then the rate-of-progress variable of the k-th reaction, q_k, is

$$q_k = k_{fk}\prod_{\mathfrak{m}\in\mathcal{M}}[\mathfrak{m}]^{\nu'_{\mathfrak{m}k}} - k_{rk}\prod_{\mathfrak{m}\in\mathcal{M}}[\mathfrak{m}]^{\nu''_{\mathfrak{m}k}} \tag{2.12}$$

where k_{fk} and k_{rk} are the forward and backward reaction rates, and are usually expressed in Arrhenius forms,

$$k_f = AT^n \exp\left(-E_A/R_{\mathrm{u}}T\right), \tag{2.13}$$

with $[\mathfrak{m}]$ the molar concentration of species $\mathfrak{m}$. The reaction index k is suppressed from this expression for conciseness, but parameters A, n, and E_A are all reaction-dependent. Each one of these parameters is usually supplied by reaction models such as GRI-Mech 3.0. Although the backward reaction rate may be obtained by the same formula, it is more accurate to use the equilibrium constant K_c to obtain K_r through (*e.g.*, Denbigh, 1955; Turns, 2000)

$$K_c = \frac{k_f}{k_r}, \tag{2.14}$$

where

$$K_c = K_p \left(\frac{p_{\mathrm{ref}}}{R_{\mathrm{u}}T}\right)^s, \tag{2.15}$$

where $s = \sum \nu^{(p)} - \sum \nu^{(r)}$, and

$$K_p = \exp\left(\sum_{\mathfrak{m}\in\mathcal{M}} \left[\nu_{\mathfrak{m}}^{(r)} \frac{g_{\mathfrak{m}}}{R_{\mathrm{u}}T} - \nu_{\mathfrak{m}}^{(p)} \frac{g_{\mathfrak{m}}}{R_{\mathrm{u}}T}\right]\right). \tag{2.16}$$

The molar production rate is

$$\dot{\omega}_{\mathfrak{m}}^c = \sum_{k=1}^{K} \left(\nu_{\mathfrak{m}k}^{(p)} - \nu_{\mathfrak{m}k}^{(r)}\right) q_k. \tag{2.17}$$

The mass fraction production rate can be obtained by

$$\dot{\omega}_{\mathfrak{m}} = \frac{W_{\mathfrak{m}}}{\rho} \dot{\omega}_{\mathfrak{m}}^c. \tag{2.18}$$

2.2.5 The mixture-averaged transport model

In addition to a chemical-kinetics model, transport properties that appear in the equations must be modeled and evaluated. Models used in the present study are described here for completeness, and more details can be found in Kee *et al.* (1986). The Wilke formula is

used to evaluate the viscosity of the given gas mixture,

$$\mu = \sum_{\mathfrak{m}\in\mathcal{M}} \frac{X_{\mathfrak{m}}\mu_{\mathfrak{m}}}{\sum_{\mathfrak{n}} X_{\mathfrak{n}}\Phi_{\mathfrak{m}\mathfrak{n}}}, \tag{2.19}$$

where

$$\Phi_{\mathfrak{m}\mathfrak{n}} = \frac{1}{\sqrt{8}}\left(1+\frac{W_{\mathfrak{m}}}{W_{\mathfrak{n}}}\right)^{-1/2}\left(1+\left(\frac{\mu_{\mathfrak{m}}}{\mu_{\mathfrak{n}}}\right)^{1/2}\left(\frac{W_{\mathfrak{n}}}{W_{\mathfrak{m}}}\right)^{1/4}\right)^{2}. \tag{2.20}$$

$X_{\mathfrak{m}}$ is the mole fraction of species $\mathfrak{m}$, $\mu_{\mathfrak{m}}$ is the dynamic viscosity of species $\mathfrak{m}$, which is given by kinetic theory,

$$\mu_{\mathfrak{m}} = \frac{5}{16}\frac{\sqrt{\pi m_{\mathfrak{m}} k_B}}{\pi\sigma_{\mathfrak{m}}^2\Omega^{(2,2)*}}, \tag{2.21}$$

where $\sigma_{\mathfrak{m}}$ is the Lennard-Jones collision diameter, $m_{\mathfrak{m}}$ is the molecular mass ($W_{\mathfrak{m}}/N_A$ where N_A is the Avogadro number), and k_B is the Boltzmann constant. The collision integral $\Omega^{(2,2)*}$ values are evaluated through the tables given in Monchick & Mason (1961).

The thermal conductivity of a gas mixture is computed using a combination averaging formula:

$$\lambda_T = \frac{1}{2}\left(\sum_m X_{\mathfrak{m}}\lambda_{T\mathfrak{m}} + \frac{1}{\sum_m X_{\mathfrak{m}}/\lambda_{T\mathfrak{m}}}\right). \tag{2.22}$$

The thermal conductivity of each individual species can be obtained by

$$\lambda_{T\mathfrak{m}} = \frac{\mu_{\mathfrak{m}}}{W_{\mathfrak{m}}}\left(f_{\text{trans}}C_{v,\text{trans}} + f_{\text{rot}}C_{v,\text{rot}} + f_{\text{vib}}C_{v,\text{vib}}\right), \tag{2.23}$$

where

$$f_{\text{trans}} = \frac{5}{2}\left(1-\frac{2}{\pi}\frac{C_{v,\text{rot}}}{C_{v,\text{trans}}}\frac{A}{B}\right), \tag{2.24}$$

$$f_{\text{rot}} = \frac{\rho D_{mm}}{\mu_{\mathfrak{m}}}\left(1+\frac{2}{\pi}\frac{A}{B}\right), \tag{2.25}$$

and

$$f_{\text{vib}} = \frac{\rho D_{mm}}{\mu_{\mathfrak{m}}}. \tag{2.26}$$

The constants that appear in these expressions are

$$A = \frac{5}{2} - f_{\text{vib}}, \tag{2.27}$$

Table 2.1: The values of C_v

	monatomic	linear	nonlinear
$C_{v,\mathrm{trans}}/R_\mathrm{u}$	3/2	3/2	3/2
$C_{v,\mathrm{rot}}/R_\mathrm{u}$	0	1	3/2
$C_{v,\mathrm{vib}}/R_\mathrm{u}$	0	$Cv/R_\mathrm{u} - 5/2$	$Cv/R_\mathrm{u} - 3$

and

$$B = Z_\mathrm{rot} + \frac{2}{\pi}\left(\frac{5}{3}\frac{C_{v,\mathrm{rot}}}{R_\mathrm{u}} + f_\mathrm{vib}\right), \tag{2.28}$$

where Z_rot is the rotational relaxation collision number, and its value at 298 K is given as an input.

The molar heat capacities in Eqn.(2.23) depend on the geometry of molecules and are collected in Table 2.1.

The self-diffusion coefficient is given by,

$$D_{\mathfrak{m}\mathfrak{m}} = \frac{3}{16}\frac{\sqrt{2\pi k_B^3 T^3/m_\mathfrak{m}}}{p_0 \pi \sigma_k^2 \Omega^{(1,1)*}}. \tag{2.29}$$

The mixture-averaged diffusion coefficient is given by Mathur *et al.* (1967) (see Kee *et al.*, 1986)

$$D_\mathfrak{m} = \frac{\bar{W} - X_\mathfrak{m} W_\mathfrak{m}}{\bar{W}\sum_{\mathfrak{n}\neq\mathfrak{m}} X_\mathfrak{n}/D_{\mathfrak{m}\mathfrak{n}}}, \tag{2.30}$$

where $D_{\mathfrak{m}\mathfrak{n}}$ is the binary diffusion coefficient between species $\mathfrak{m}$ and $\mathfrak{n}$,

$$D_{\mathfrak{m}\mathfrak{n}} = \frac{3}{16}\frac{\sqrt{2\pi k_B^3 T^3/m_{\mathfrak{m}\mathfrak{n}}}}{p\pi\sigma_{\mathfrak{m}\mathfrak{n}}\Omega^{(1,1)*}}, \tag{2.31}$$

where $m_{\mathfrak{m}\mathfrak{n}}$ is the reduced molar mass for the species pair and is given by

$$m_{\mathfrak{m}\mathfrak{n}} = \frac{m_\mathfrak{m} m_\mathfrak{n}}{m_\mathfrak{m} + m_\mathfrak{n}}. \tag{2.32}$$

2.3 The numerical method

2.3.1 Introduction

There are many methods for solving partial differential equations numerically. However, for the purpose of this study, the requirements are: flexibility to handle complex geometry as set up in a laboratory, efficiency to make it possible to integrate computationally expensive reacting flow equations, and accuracy to manage numerical error so that the errors in the chemical kinetics models can be evaluated and assessed. The spectral element method, originally proposed by Patera (1984) for incompressible flows (*e.g.*, Henderson & Karniadakis, 1995; Henderson & Barkley, 1996; Henderson, 1999a,b; Blackburn & Henderson, 1999; Tomboulides & Orszag, 2000; Blackburn & Lopez, 2002), can be adapted to satisfy all these requirements after its applicability is extended to accommodate flows with large density variations. Although the original method was intended for an incompressible flow, Tomboulides *et al.* (1997) and Tomboulides & Orszag (1998) later extended it to variable-density low Mach number flow. However their approach is still not sufficiently efficient to simulate methane flames with detailed combustion and transport models because their algorithm solves the reaction-diffusion equations without operator splitting. A new algorithm has been developed and is presented here.

2.3.2 Expansion basis

In the spectral element method for two-dimensional problems, the approximate solution is expanded in a given expansion basis as follows,

$$u(x,y) = \sum_{i,j} u_{i,j} h_i(x) h_j(y), \tag{2.33}$$

where $h_i(x)$ is a compactly supported Lagrange polynomial based on the roots of Jacobi polynomials and has the following properties:

$$h_i(x_j) = \delta_{ij}, \tag{2.34}$$

$$\int_{\Omega^e} u(x)dx \approx \sum_i \omega_i u(x_i), \tag{2.35}$$

and

$$\int_{\Omega^e} h_i(x)h_j(x)dx \approx \begin{cases} \omega_i & i = j \\ 0 & i \neq j \end{cases}, \tag{2.36}$$

where Ω^e is a finite size domain over which the basis functions, $h_i(x)$, have support.

The basis function $h_i(x)$ can be constructed using many different orthogonal polynomials, but Legendre polynomials with Gauss-Lobatto quadrature are most often used (Henderson & Karniadakis, 1995; Blackburn & Sherwin, 2004), and they will also be used in this study (hereafter called GLL—Gauss-Lobatto-Legendre—basis) except for the expansion in the radial direction within elements that are adjacent to the axis. For this special case, a new basis function has been developed and will be described next.

2.3.3 Polar axis treatment for axisymmetric flow

Simulation of fluid flow in an axisymmetric domain requires a proper treatment of the axis singularity. There are essentially two issues: One is that the governing equation itself contains a singularity when written in polar-coordinate form, and the other is that there are certain requirements in the behavior of each azimuthal Fourier mode, and those requirements must be satisfied by each expansion basis function. These issues are addressed in the context of the spectral method by several authors (Leonard & Wray, 1982; Matsushima & Marcus, 1995; Mohseni & Colonius, 2000) (see Boyd, 2000), but the second issue has been mostly ignored in the spectral element method (Tomboulides *et al.*, 1997; Blackburn & Sherwin, 2004). In those studies that employed the spectral element method in axisymmetric coordinates, only the leading order behavior is specified. Higher-order conditions are ignored with the expectation that they will be satisfied in the process of convergence. However, although there are cases in which high-order conditions can be satisfied without specifying them, there are cases in which the smooth numerical solution does not satisfy some particular required property. To this end, a new basis function that incorporates the correct behavior has been developed.

An arbitrary function in polar coordinates can be expanded as a Fourier series in θ,

$$f(r,\theta) = \sum_{m=0}^{\infty} \left(f_m(r)\cos(m\theta) + g_m(r)\sin(m\theta) \right). \tag{2.37}$$

If $f(r,\theta)$ is a scalar and the function is analytic at $r = 0$, the following conditions must be

satisfied (Boyd, 2000):

(i) $f_m(r)$ and $g_m(r)$ have m-th order zeros at $r = 0$.

(ii) If m is even, then $f_m(r)$ and $g_m(r)$ are both symmetric about $r = 0$ and their power series contain only even powers of r.

(iii) If m is odd, then $f_m(r)$ and $g_m(r)$ are both antisymmetric about $r = 0$ and their power series contain only odd powers of r.

When $f(r, \theta)$ is the axial velocity in the cylindrical coordinates or is the product of r with the radial or tangential velocity in polar coordinates, the same conditions apply. For axisymmetric problems, as considered in this paper, the requirement translates to the fact that scalars and axial velocity must be even functions about the axis, while radial velocity must be an odd function about the axis, and we need to implement this property in the basis function itself.

In the standard coordinate, $\xi \in [-1, 1]$, mapping to the physical coordinate is provided by isoparametric mapping (Karniadakis & Sherwin, 1999). The approximate solution is expanded in the following form:

$$f(\xi_z, \xi_r) = \sum_{i=0}^{Q-1} \sum_{j=0}^{Q-1} \hat{f}_{ij} h_i(\xi_z) h_j^r(\xi_r) \tag{2.38}$$

where $h_i(z)$ are Lagrange polynomials of order P $(= Q - 1)$. For the expansion of all elements in the axial direction and for the expansion in the radial direction (except for those elements adjacent to the axis), GLL collocation points are used to construct the Lagrange polynomials such that $h_m(\xi_n) = \delta_{mn}$, where $\xi_m (m = 0, 1, \ldots, Q - 1)$ are a set of Q GLL points. This is a standard basis in the spectral element method and further details are given in Karniadakis & Sherwin (1999). For $h_m^r(\xi)$, depending on the parity requirements for the approximate solution, either $h_m^{\text{even}}(\xi)$ or $h_m^{\text{odd}}(\xi)$ are used—which are even and odd functions of ξ, respectively—or $h_m(\xi)$ is used if there is no parity requirement.

For the radial expansion of elements adjacent to the axis, let $\eta = [(\xi + 1)/2]^{1/2} \in [0, 1]$. Following Leonard & Wray (1982), and Matsushima & Marcus (1995), a function with even parity about $\eta = 0$ can be expressed as

$$h_m^{\text{even}}(\eta) = h_m(2\eta^2 - 1). \tag{2.39}$$

Leonard & Wray (1982) used a quadratic basis function to obtain the desired parity and the divergence-free constraint in their numerical study using a spectral method. Later, Matsushima & Marcus (1995) derived a more general form of basis functions with parity that can represent any non-symmetric smooth function with the required parity property. The basis function used by Leonard & Wray (1982) is a special case of the one derived by Matsushima & Marcus (1995).

Effectively, we construct a Lagrange polynomial using quadratic monomials (x^2) to obtain a desired parity, and a required matching condition at the outer boundary. However, this is still a GLL approximation in the original polynomial space, and we can achieve spectral accuracy, as shown later. Since $\{L_m(x)\}_{m=0,\ldots,Q-1}$ is a complete basis for a space of polynomials of order P, this basis function spans a space of even polynomials of order $2P$. The new basis functions are not orthogonal analytically (so are the original GLL polynomials), but they both satisfy the discrete orthogonality conditions.

The basis function $h_m(\eta)$ is an even function in η and $h_m(\eta_n) = \delta_{mn}$ so that the discrete orthogonality condition is satisfied.

For the odd function basis, one may use $h_m^{\text{even}}(\eta)$ as a building block, and then

$$h_m^{\text{odd}}(\eta) = \frac{\eta}{\eta_m} h_m^{\text{even}}(\eta) \tag{2.40}$$

satisfies all the requirements.

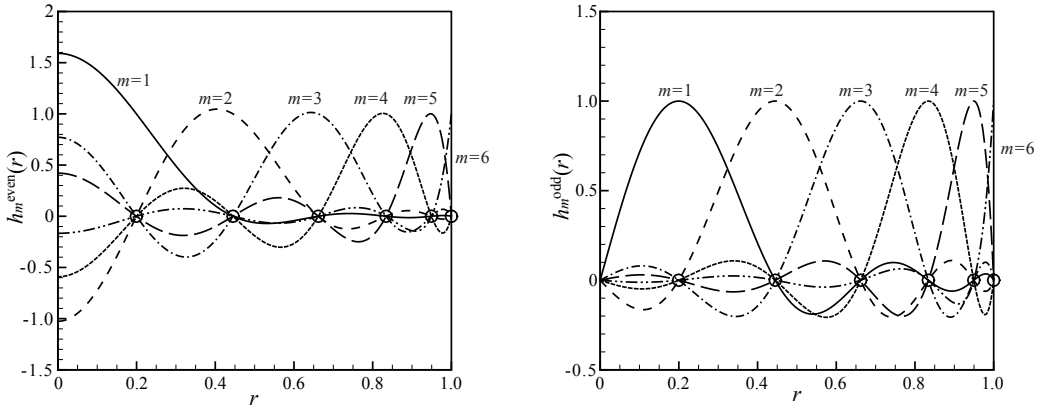
Simple arithmetic shows that the collocation derivative matrices for these bases are

$$D_{ij}^{\text{even}} = 4\frac{r_i}{R^2} D_{ij}, \tag{2.41}$$

and

$$D_{ij}^{\text{odd}} = \frac{1}{r_i}\delta_{ij} + \frac{r_i}{r_j} D_{ij}^{\text{even}} \tag{2.42}$$

where D_{ij} is the collocation differentiation matrix for the original basis, and R is the length of the radial domain in physical space. The even and odd basis functions can be constructed through Eqns.(2.39) and (2.40) once the base basis function, h_m in Eqn.(2.39) is selected. The Gauss-Radau-Legendre (GRL) polynomial has the desired property of containing only one boundary collocation point so that it allows connectivity conditions to be incorporated at the outer element boundary while it spans all polynomial space less than the given



(a) Even function basis ($Q = 6$)

(b) Odd function basis ($Q = 6$)

Figure 2.1: The radial expansion bases for elements adjacent to the axis. Note that each basis function satisfies the correct parity requirements. Each basis function consists of up to $(2Q-1)$-th order polynomials and satisfies the discrete orthogonality property.

polynomial order. In addition, the GRL basis conveniently avoids evaluating singular terms in the governing equation by avoiding a collocation point on the axis.

The Gaussian quadrature weights are $w_i^{\text{even}} = w^{\text{odd}} = (R^2/4)w_i$. These new bases are shown in Fig. 2.1, for the case of $Q = 6$, and they will be referred to as Gauss-Radau-Legendre with parity (GRLp) bases in this study. As can be seen in the figure, the collocation points associated with this basis function have fewer clustered collocation points toward the axis, where the function expanded is expected to be smooth. Therefore, it is slightly computatinally advantageous compared to other basis functions used in earlier works on spectral element method for axisymmetric problems (Tomboulides *et al.*, 1997; Blackburn & Sherwin, 2004).

$$
\begin{aligned}
&\int_0^R f^{\text{even}}(r) r dr \\
&= R^2 \int_0^1 \sum_m f_m h_m^{\text{even}}(\eta) \eta d\eta \\
&= R^2 \sum_m f_m \int_0^1 h_m^{\text{even}}(\eta) \eta d\eta \\
&= \frac{R^2}{4} \sum_m f_m \int_{-1}^1 h_m(\xi) d\xi \\
&= \frac{R^2}{4} \sum_m f_m \left(\sum_n w_n h_m(\xi_n) \right) \\
&= \frac{R^2}{4} \sum_m f_m w_m \\
&= \sum_m f_m w_m^{\text{even}}.
\end{aligned}
$$

The same is true for the odd basis:

$$
\begin{aligned}
&\int_0^R f^{\text{odd}}(r) r dr \\
&= R^2 \int_0^1 \sum_m f_m h_m^{\text{odd}}(\eta) \eta d\eta \\
&= R^2 \sum_m f_m \int_0^1 h_m^{\text{odd}}(\eta) \eta d\eta \\
&= R^2 \sum_m f_m \int_0^1 \frac{\eta}{\eta_m} h_m^{\text{even}}(\eta) \eta d\eta \\
&= \frac{R^2}{4} \sum_m f_m \int_{-1}^1 \frac{\eta(\xi)}{\eta_m} h_m(\xi) d\xi \\
&= \frac{R^2}{4} \sum_m f_m \left(\sum_n w_n \frac{\eta_n}{\eta_m} h_m(\xi_n) \right) \\
&= \frac{R^2}{4} \sum_m f_m w_m \\
&= \sum_m f_m w_m^{\text{even}}.
\end{aligned}
$$

Development of basis functions with appropriate parity and desired behavior for the non-axisymmetric case, *i.e.*, $m > 0$, near the axis is a straightforward extension of the approach described here and the work of Matsushima & Marcus (1995), using one-sided

Table 2.2: The mixed stiffly stable scheme coefficients (Karniadakis *et al.*, 1991)

Coefficient	γ_0	α_0	α_1	α_2	β_0	β_1	β_2
1st order	1	1	0	0	1	0	0
2nd order	3/2	2	-1/2	0	2	-1	0
3rd order	11/6	3	-3/2	1/3	3	-3	1

Jacobi polynomials.

2.3.4 The spectral element method for uniform-density flows

We first introduce the spectral element method for uniform-density flow, as we will use the same building blocks later to construct a method for reacting flows.

First, the momentum equations are discretized in time using a mixed stiffly stable scheme (Karniadakis *et al.*, 1991). The idea of a mixed stiffly stable scheme is to use an extrapolation to estimate the implicit flux of the nonlinear term in the context of a stable backward-differentiation scheme.

$$\frac{\gamma_0 \mathbf{u}^{n+1} - \sum_{q=0}^{J_e-1} \alpha_q \mathbf{u}^{n-q}}{\Delta t} = \sum_{q=0}^{J_e-1} \beta_q \mathbf{N}(\mathbf{u}^{n-q}) - \nabla p^{n+1} + \nu \mathbf{L}(\mathbf{u}^{n+1}) \tag{2.43}$$

where $\mathbf{N}(\mathbf{u}) = -\mathbf{u} \cdot \nabla \mathbf{u}$, and $\mathbf{L}(\mathbf{u}) = \nabla^2 \mathbf{u}$. In the above equations, α_q, β_q, and γ_0 are the weighting coefficients for stiffly stable time-integration method of order J_e, and the values are tabulated in Table 2.2.

Splitting this equation into terms gives,

$$\widehat{\mathbf{u}} - \sum_{q=0}^{J_e-1} \alpha_q \mathbf{u}^{n-q} = -\Delta t \sum_{q=0}^{J_e-1} \beta_q \left(\mathbf{u} \cdot \nabla \mathbf{u}\right)^{n-q} \tag{2.44a}$$

$$\widehat{\widehat{\mathbf{u}}} - \widehat{\mathbf{u}} = -\Delta t \nabla p^{n+1} \tag{2.44b}$$

$$\gamma_0 \mathbf{u}^{n+1} - \widehat{\widehat{\mathbf{u}}} = \nu \Delta t \nabla^2 \mathbf{u}^{n+1}, \tag{2.44c}$$

where $\widehat{\mathbf{u}}$ and $\widehat{\widehat{\mathbf{u}}}$ are intermediate velocities in the time-stepping scheme, defined by Eqns.(2.44a) and (2.44b), respectively.

By taking the divergence of the equation for the pressure projection, Eqn. (2.44b), the pressure Poisson equation is obtained. Note that $\nabla \cdot \widehat{\widehat{\mathbf{u}}} = 0$ by taking the divergence of the

last equation. Finally, the governing equations are integrated by the following four steps:

$$\widehat{\mathbf{u}} - \sum_{q=0}^{J_e-1} \alpha_q \mathbf{u}^{n-q} = -\Delta t \sum_{q=0}^{J_e-1} \beta_q \left(\mathbf{u} \cdot \nabla \mathbf{u}\right)^{n-q} \tag{2.45a}$$

$$\nabla^2 p^{n+1} = \frac{1}{\Delta t} \nabla \cdot \widehat{\mathbf{u}} \tag{2.45b}$$

$$\widehat{\widehat{\mathbf{u}}} - \widehat{\mathbf{u}} = -\nabla p^{n+1} \Delta t \tag{2.45c}$$

$$\gamma_0 \mathbf{u}^{n+1} - \widehat{\widehat{\mathbf{u}}} = \nu \Delta t \nabla^2 \mathbf{u}^{n+1}. \tag{2.45d}$$

It may be observed that the building blocks for the spectral element method for incompressible uniform density flows are the explicit advection equation,

$$\frac{\partial u}{\partial t} = f(u), \tag{2.46}$$

and the implicit Helmholtz equation,

$$\nabla^2 u - \lambda u = f(u). \tag{2.47}$$

The former is solved by an explicit collocation method while the latter is solved by the Galerkin method.

2.3.5 The spectral element method for variable-density flows

When extending the spectral element method to the chemically reacting flows with large density variations, additional difficulties arise.

First, unlike in the compressible counterpart, pressure must be obtained by deriving and solving the pressure Poisson equation rather than the equation of state. This is because pressure in the equation of state is not the same as that in the momentum equation. Second, when solving the compressible equations explicitly, the continuity equation can be omitted in favor of the species-conservation equations and the constraint, $\sum_{\mathfrak{m}} Y_{\mathfrak{m}} = 1$. Then the density field can be obtained simply by adding partial densities, *i.e.*, $\rho = \sum_{\mathfrak{m}} \rho_{\mathfrak{m}}$. Again, this is not possible in the low Mach number case as the continuity equation is needed to derive the pressure Poisson equation, and therefore, one of the species conservation equations is redundant and must be abandoned. Third, the equation of state is now a constraint between

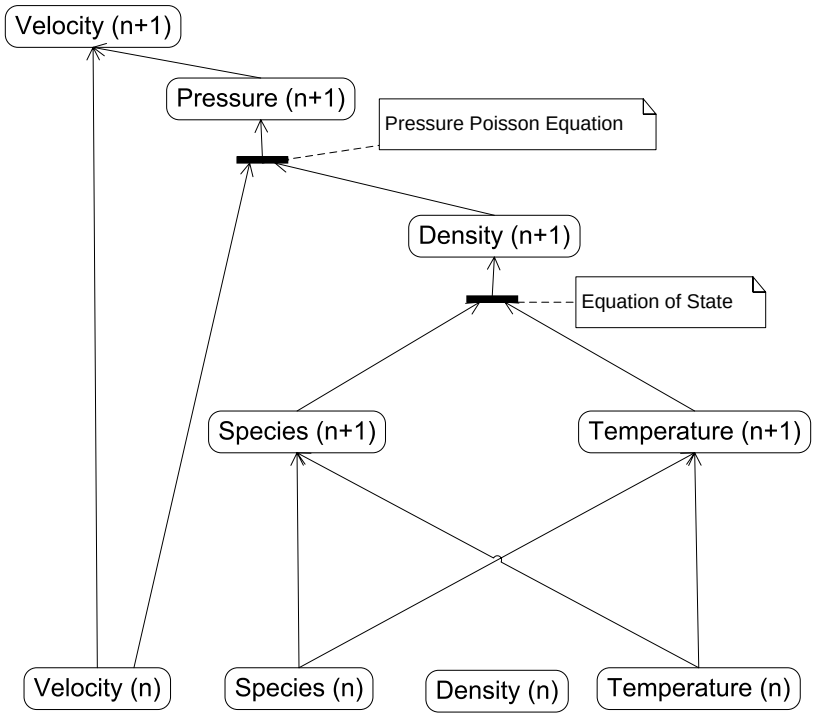


Figure 2.2: Integration dependency for low Mach number formulation

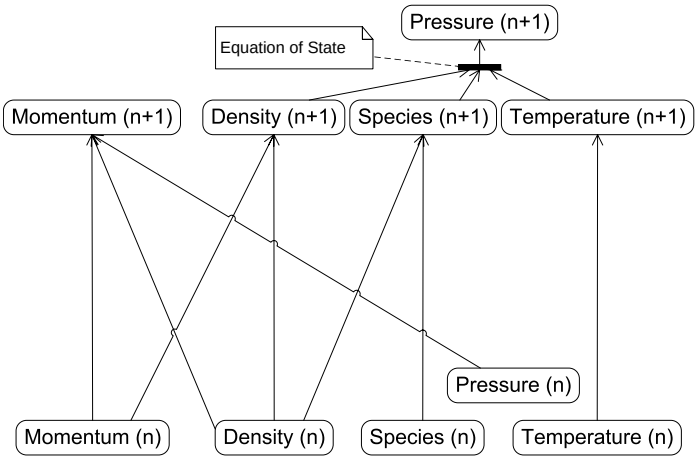


Figure 2.3: Integration dependency for explicit compressible formulation

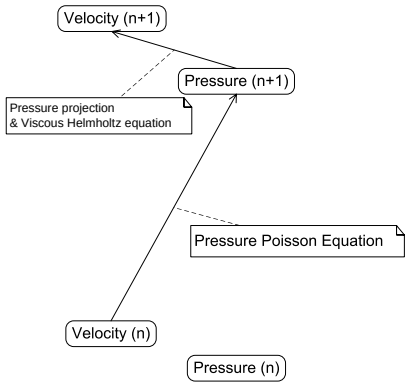


Figure 2.4: Integration dependency for uniform density formulation

density, compositions through the specific gas constant, and temperature—and among these unknowns, only density does not have a formula that updates it in time. Therefore, the equation of state must be used to specify the density field, given the temperature and mass fractions. This is a major difference from the work of Day & Bell (2000), in which the continuity equation was used to update density while the equation of state is incorporated

in the divergence constraint as a penalty barrier.

The updating scheme is summarized in Fig. 2.2. For comparison, the same dependency diagrams for a compressible formulation (Fig. 2.3) and a uniform-density formulation (Fig. 2.4) are shown here as well. As can be seen from this diagram, a density update must be obtained before integrating the momentum equations. Therefore, temperature and species must be updated first.

In addition, the left hand side operator of the pressure Poisson equation now contains the reciprocal of the density, $1/\rho$, and a technique for solving such equations must be developed. One could choose not to include this $1/\rho$ factor inside the operator by multiplying through ρ to the momentum equations before taking the divergence of the entire equation. However, this formulation limits the size of the timestep, Δt, compared to the current formulation and was not used in this study.

The discrete form of the governing equations is introduced in the order in which they are updated:

$$\frac{\gamma_0 T^{n+1} - \sum_{q=0}^{J_e-1} \alpha_q T^{n-q}}{\Delta t} = \sum_{q=0}^{J_e-1} \beta_q \left(-\mathbf{u}\cdot\nabla T + \left(\frac{1}{\rho C_{p,\mathrm{mix}}}\right)\nabla\lambda_T\cdot\nabla T\right)^{n-q} + \alpha^n \nabla^2 T^{n+1} \tag{2.48}$$

$$\frac{\gamma_0 Y_{\mathfrak{m}}^{n+1} - \sum_{q=0}^{J_e-1} \alpha_q Y_{\mathfrak{m}}^{n-q}}{\Delta t} = \sum_{q=0}^{J_e-1} \beta_q \left(-\mathbf{u}\cdot\nabla Y_{\mathfrak{m}} + \frac{1}{\rho^n}\nabla\rho D_{\mathfrak{m}}\cdot\nabla Y_{\mathfrak{m}}\right)^{n-q} + D_{\mathfrak{m}}^n \nabla^2 Y_{\mathfrak{m}}^{n+1} \tag{2.49}$$

for $\mathfrak{m} \in \mathcal{M}'$ where $\mathcal{M}'$ contains all species in $\mathcal{M}$ except N_2. Then,

$$Y_{N_2} = 1 - \sum_{\mathfrak{m}\in\mathcal{M}'} Y_{\mathfrak{m}}, \tag{2.50}$$

and

$$\rho^{n+1} = \frac{p_0}{R_u T^{n+1}}\bar{W} = \frac{p_0}{R_u T^{n+1}} \frac{1}{\sum_{\mathfrak{m}\in\mathcal{M}}(Y_{\mathfrak{m}}^{n+1}/W_{\mathfrak{m}})}. \tag{2.51}$$

Once the updated density, temperature, and mass fractions are obtained, the rest follows the procedure for incompressible non-uniform density flows, which will be described next. Now that species and temperature fields are updated, transport properties can be updated

at this time, *i.e.*,

$$\mu^{n+1} = \mu(Y_{\mathfrak{m}}^{n+1}, T^{n+1}) \tag{2.52a}$$

$$\lambda_T^{n+1} = \lambda_T(Y_{\mathfrak{m}}^{n+1}, T^{n+1}) \tag{2.52b}$$

$$D_{\mathfrak{m}}^{n+1} = D_{\mathfrak{m}}(Y_{\mathfrak{m}}^{n+1}, T^{n+1}, p_0) \tag{2.52c}$$

Once all thermodynamic states at the new time-level are obtained, the momentum equations are finally integrated,

$$\widehat{\mathbf{u}} - \sum_{q=0}^{J_e-1} \alpha_q \mathbf{u}^{n-q} = \Delta t \sum_{q=0}^{J_e-1} \alpha_q \left(-\mathbf{u}\cdot\nabla\mathbf{u} + \mathbf{L}^e(\mathbf{u}) + \mathbf{f}\right)^{n-q} \tag{2.53a}$$

$$\nabla\cdot\left(\frac{1}{\rho}\nabla p^*\right) = \frac{1}{\Delta t}\left(\nabla\cdot\widehat{\mathbf{u}} - \nabla\cdot\widehat{\widehat{\mathbf{u}}}\right) \tag{2.53b}$$

$$\widehat{\widehat{\mathbf{u}}} - \widehat{\mathbf{u}} = -\Delta t\,\nabla p^* \tag{2.53c}$$

$$\gamma_0\mathbf{u}^{n+1} - \widehat{\widehat{\mathbf{u}}} = \frac{\mu}{\rho}\Delta t\,\nabla^2\mathbf{u}^{n+1}, \tag{2.53d}$$

where, again as is in the uniform-density equations, $\widehat{\mathbf{u}}$ and $\widehat{\widehat{\mathbf{u}}}$ are the intermediate velocities in the time-stepping scheme, defined by Eqns.(2.53a) and (2.53c), respectively.

In the above expressions, J_e in summations is the order of the time-integration scheme, and appropriate coefficients (α_q, β_q, and γ_0) for each case are recorded in Table 2.2. In contrast with the uniform-density case, the second term on the right-hand side of the pressure Poisson equation does not disappear. Now, to obtain the unknown quantity that appears in the right hand side of Eqn.(2.53b), we use Eqn.(2.53d),

$$\nabla\cdot\widehat{\widehat{\mathbf{u}}} = \gamma_0 Q^{n+1} - \Delta t\nabla\cdot\left[\frac{\mu}{\rho}\Big(\nabla\left(\nabla\cdot\mathbf{u}\right) - \nabla\times\nabla\times\mathbf{u}\Big)\right], \tag{2.54}$$

where Q^{n+1} is the divergence constraint that must be satisfied at time t^{n+1}, which must be obtained through the continuity equation,

$$Q^{n+1} \equiv \nabla\cdot\mathbf{u}^{n+1} = -\frac{1}{\rho}\frac{D\rho}{Dt} = -\frac{1}{\rho}\left(\frac{\gamma_0\rho^{n+1} - \sum_{q=0}^{J_e-1}\alpha_q\rho^{n-q}}{\Delta t} + \sum_{q=0}^{J_e-1}\beta_q(\mathbf{u}\cdot\nabla\rho)^{n-q}\right). \tag{2.55}$$

Other studies (Tomboulides *et al.*, 1997; Day & Bell, 2000) used the Lagrange derivative of the equation of state to decompose $D\rho/Dt$ into DT/Dt and $DY_{\mathfrak{m}}/Dt$, and used the tem-

perature transport equation and the species transport equations to obtain Q^{n+1}. However this approach is costly and accumulates round-off errors.

2.3.6 Boundary condition for the pressure Poisson equation

The pressure Poisson equation requires boundary conditions. Typically, uniform-pressure boundary condition is specified at the outflow where the ambient pressure can be specified. For other boundaries, such as inflow and wall, where the velocity components are specified, the Neumann boundary condition for the pressure field must be derived from the momentum equation. Assuming the steady boundary condition, we have,

$$\frac{1}{\rho}\frac{\partial p^*}{\partial n} = \mathbf{n}\cdot\left(\mathbf{N}(\mathbf{u}) + \mathbf{L}(\mathbf{u})\right), \tag{2.56}$$

where $\mathbf{N}(\mathbf{u}) = -\mathbf{u}\cdot\nabla\mathbf{u}$.

Using the vector identity,

$$\nabla^2\mathbf{u} = \nabla\left(\nabla\cdot\mathbf{u}\right) - \nabla\times\nabla\times\mathbf{u}, \tag{2.57}$$

the viscous term can be rewritten as,

$$\mathbf{L}(\mathbf{u}) = \frac{\mu}{\rho}\left(2\nabla Q - \nabla\times\nabla\times\mathbf{u}\right) + \frac{2}{\rho}\nabla\mu\cdot\mathbf{S}, \tag{2.58}$$

where $\mathbf{S} = (\nabla\mathbf{u} + \nabla\mathbf{u}^T)/2$.

2.3.7 Helmholtz equations

Solving Eqns. (2.48), (2.49), (2.53b), and (2.53d) requires solving the Helmholtz equations. We define the inner-product to describe Galerkin formulation,

$$(\psi, \Phi) = \int_\Omega \psi(\mathbf{x})\Phi(\mathbf{x})d\Omega, \tag{2.59}$$

$$(\psi, \Phi)_w = \int_\Omega \psi(\mathbf{x})\Phi(\mathbf{x})w(\mathbf{x})d\Omega, \tag{2.60}$$

and

$$\langle\psi, \Phi\rangle = \int_{\partial\Omega^{\mathcal{N}}} \psi(\mathbf{x})\nabla\Phi(\mathbf{x})\cdot\mathbf{n}dS. \tag{2.61}$$

2.3.7.1 Cartesian coordinate

In this case,

$$\nabla^2 u - \lambda^2 u = f(u), \tag{2.62}$$

or in differential form,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - \lambda u = f(u). \tag{2.63}$$

We multiply the equation by a test function $v(x, y)$ and integrate over the entire domain to obtain,

$$(\nabla v, \nabla u) + \lambda(v, u) = -(v, f) + \langle v, u \rangle. \tag{2.64}$$

Using the basis functions described earlier, this equation can be cast into the linear system.

$$[\mathbf{L} + \lambda \mathbf{M}]\, \mathbf{u} = -\mathbf{M}\mathbf{f} + \mathbf{u}^{\delta N} \tag{2.65}$$

where

$$L_{ij} = \int_\Omega h_i'(\mathbf{x}) h_j'(\mathbf{x}) d\Omega. \tag{2.66}$$

2.3.7.2 Cylindrical coordinate

Again, we start with the same equation,

$$\nabla^2 u - \lambda^2 u = f(u), \tag{2.67}$$

which in differential form is,

$$\frac{\partial^2 u}{\partial z^2} + \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) - \lambda^2 u = f(u). \tag{2.68}$$

We multiply this equation through by r to obtain (Blackburn & Sherwin, 2004),

$$\frac{\partial}{\partial z}\left(r\frac{\partial u}{\partial z}\right) + \frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) - \lambda^2 r u = r f(u). \tag{2.69}$$

Again we apply the method of weighted residuals with a test function $v(z, r)$,

$$(\nabla v, \nabla u)_r + \lambda^2 (v, u)_r = -(v, f)_r + \langle v, u \rangle_r. \tag{2.70}$$

As it appears in the pressure Poisson equation, an equation with a variable factor inside the operator needs to be solved:

$$\nabla \cdot \left(\frac{1}{\rho} \nabla u \right) - \lambda^2 u = f(u), \tag{2.71}$$

which in differential form is,

$$\frac{\partial}{\partial z} \left(\frac{r}{\rho} \frac{\partial u}{\partial z} \right) + \frac{\partial}{\partial r} \left(\frac{r}{\rho} \frac{\partial u}{\partial r} \right) - \lambda^2 r u = r f(u). \tag{2.72}$$

Again, the factor r has been applied to the entire equation. Applying the method of weighted residuals with a test function $v(z, r)$ then yields,

$$(\nabla v, \nabla u)_{r/\rho} + \lambda^2 (v, u)_r = -(v, f)_r + \langle v, u \rangle_r. \tag{2.73}$$

In the exact Galarkin formulation, the weight term must be evaluated by expansion, *i.e.*, the stiffness matrix should be:

$$\mathbf{L}_{ij} = \int_{\Omega^e} \frac{\phi_i'(x)\phi_j'(x)}{\sum_{l=1}^{Q} \rho_l \phi_l(x)} d\Omega^e.$$

However, using the property of the Lagrange polynomial, we have,

$$\mathbf{L}_{ij} = \sum_{l=1}^{Q} \omega_l \frac{\phi_i'(x_l)\phi_j'(x_l)}{\rho_l}.$$

This form of the stiffness matrix is used in this study.

2.3.8 Solution method for the linear system

Now that a matrix system has been obtained for the viscous Helmholtz equations, or pressure Poisson equations, the linear system may be solved directly, or iteratively. In the spectral element method, the stiffness matrix usually has a localized structure and the static condensation technique is the most efficient technique for solving the equation.

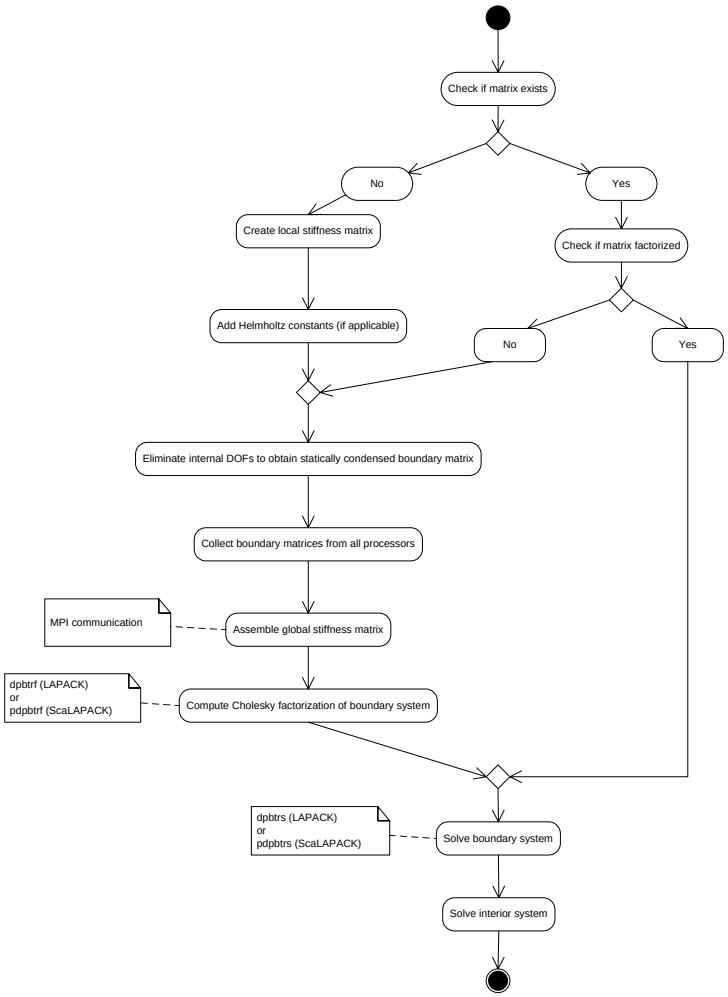


Figure 2.5: The activity diagram (flow chart) of the stiffness matrix construction and solution. Note that the Cholesky factorization is the most expensive process.

Given a matrix equation, $\mathbf{Au} = \mathbf{f}$, decompose it into a block form,

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}. \tag{2.74}$$

By multiplying

$$\begin{bmatrix} I & -A_{12}A_{22}^{-1} \\ 0 & I \end{bmatrix} \tag{2.75}$$

from the left, this equation can be factored into the block triangle form:

$$\begin{bmatrix} A_{11}^* & 0 \\ A_{21} & A_{22} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} f_1 - A_{12}A_{22}^{-1}f_2 \\ f_2 \end{pmatrix} \tag{2.76}$$

where $A_{11}^* = A_{11} - A_{12}A_{22}^{-1}A_{21}$, which is known as the Schur complement (Karniadakis & Sherwin, 1999).

By assigning the boundary degrees of freedom to u_1 and the internal degrees of freedom to u_2, the boundary (shared) degrees of freedom can be solved first, then the rest forms a set of K independent matrix equations, where K is the number of elements. The matrix A_{11}^* is a symmetric banded matrix by construction whereas the matrix A_{22} is a block-structured matrix, each of which is a positive definite matrix.

2.3.9 Computationally efficient integration of the diffusion terms

Parallelization of the algorithms and their efficient implementation is crucial in obtaining numerical solution of reacting flows. The static condensation technique described in the previous section allows a natural decomposition of the entire degrees of freedom into those on the boundary of elements and those not on the boundary. Since the interior degrees of freedom are fully decompled from the boundary degrees of freedom, the interior points can be solved with perfect scalability up to the number of elements in the computational domain. On the other hand, the matrix for the boundary degrees of freedom (the Schur complement, A_{11}^* in Eqn. 2.76) couples all degrees of freedom on element boundaries as well as domain boundaries. This poses a challenge in solving the Schur matrix efficiently, particulaly using message-passing techniques. Since, for the spectral element method, the Schur complement is usually a symmetric positive definite banded matrix, this matrix can

be solved by either an iterative method, such as conjugate gradient method, or a direct method, such as Gaussian elimination. Fischer *et al.* (1988) implemented a parallel iterative technique using a conjugate gradient and multigrid method. Subsequent work is documented in Fischer & Patera (1991), Fischer & Rønquist (1994), and Deville *et al.* (2002). However, for reacting flows, the condition number of the matrix can be significantly larger than that for incompressible flow due to large density variations, and therefore the direct method is used in the present work. `dpbtrf` and `dpbtrs` in LAPACK or `pdpbtrf` and `pdpbtrs` in ScaLAPACK libraries, both of which implement Cholesky factorization and backward substitution, can be used for this purpose. The Cholesky factorization is the most expensive operation when solving the matrix Eqn. (2.76) among the processes shown in Fig. 2.5. To gain further computational efficiency, Eqns.(2.48) and (2.49) are rewritten to improve performance.

$$\frac{T^{n+1}-T^n}{\Delta t} = \left(-\mathbf{u}\cdot\nabla T + \left(\frac{1}{\rho C_{p,\mathrm{mix}}}\right)\nabla\lambda_T\cdot\nabla T + (\alpha^n-\alpha^0)\nabla^2 T\right)^n + \alpha^0\nabla^2 T^{n+1} \quad (2.77)$$

and

$$\frac{Y_{\mathfrak{m}}^{n+1}-Y_{\mathfrak{m}}^n}{\Delta t} = \left(-\mathbf{u}\cdot\nabla Y_{\mathfrak{m}} + \frac{1}{\rho^n}\nabla\rho D_{\mathfrak{m}}\cdot\nabla Y_{\mathfrak{m}} + (D_{\mathfrak{m}}^n - D_{\mathfrak{m}}^0)\nabla^2 Y_{\mathfrak{m}}\right)^n + D_{\mathfrak{m}}^0\nabla^2 Y_{\mathfrak{m}}^{n+1}, \quad (2.78)$$

where α^0 and $D_{\mathfrak{m}}^0$ are the initial values of the thermal diffusion coefficient and the diffusion coefficient of each species, respectively.

By factoring out the time-dependent part of the diffusion operator, factorization of the matrix is required only once at the beginning of iterations. $\|\alpha^n-\alpha^0\|$ (or $\|D_{\mathfrak{m}}^n - D_{\mathfrak{m}}^0\|$) is computed at every timestep, and when it exceeds a predetermined criterion, α^0 (or $D_{\mathfrak{m}}^0$) is updated and the matrix is factorized again.

Fig. 2.6 shows execution speed per iteration per number of collocation points for reacting flow. Without the efficient method presented here, the code experiences a slowdown in execution speed after $n_p = 4$, when the ScaLAPACK library (Blackford *et al.*, 1997) is not used. When ScaLAPACK is used, it shows good scalability; however it shows a significant overhead and overall, the code does not run any faster than when ScaLAPACK is not used. When the efficient method described here is used, the code scales well both with and without

the ScaLAPACK library. In particular, use of ScaLAPACK almost halves execution time.

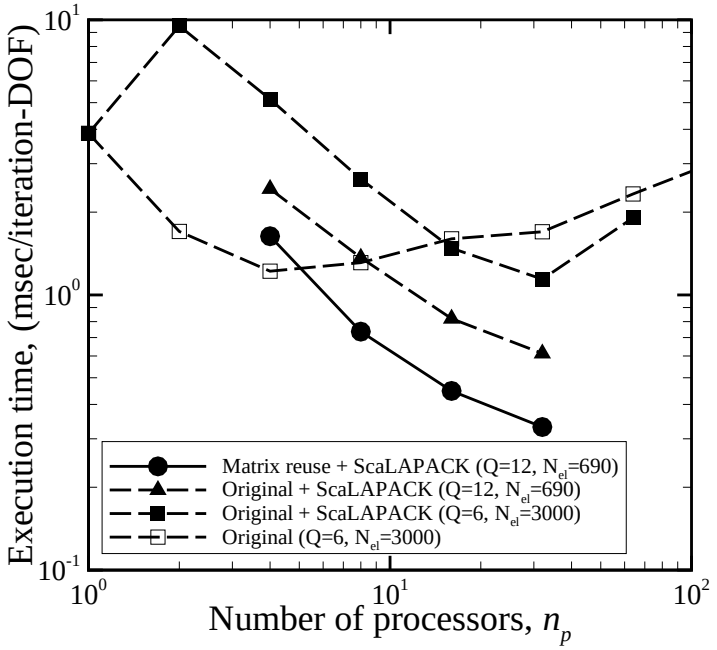


Figure 2.6: Computational performance of the Omega code used in the present study. Without the efficient integration, the code does not scale beyond $n_p = 4$. When the new method is employed, along with high-order polynomial basis, the code scales well up to $n_p = 16$ and speed up can be observed until $n_p = 32$. Use of ScaLAPACK enables the code to execute significantly faster. The benchmark test is one of Phase II simulations reported in Chapter 4. The execution time was measured on SHC cluster at the Caltech Center for Advanced Computing Research (CACR) with the code compiled with pathscale compilers.

Chapter 3

Software implementation, verification, and validation

An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software environment and the complete set of instructions which generated the figures.—D. Donoho

3.1 Introduction

In the field of computational science, it is important to have an accurate simulation environment so that we can be confident of the results and can actually rely on them in designing new products or in building better experiments.

As briefly mentioned in Chapter 1, care must be taken to ensure the correctness of the result obtained by computations. There are three kinds of error in every numerical study: modeling, numerical, and human errors.

$$\epsilon_{\text{total}} = f_{\text{error}}(\epsilon_{\text{modeling}}, \epsilon_{\text{numerical}}, \epsilon_{\text{human}}) \tag{3.1}$$

By driving these three error sources to zero, the total error should converge to zero, and the process by which the errors are small enough so that numerical solutions can be trusted is called verification or validation, depending on the type of error and the process and information employed.

The modeling error is an error in the governing equation that describes the physical system. For example, the uniform-density low Mach number flow is well described by the incompressible Navier-Stokes equations, *i.e.*, Eqn. (2.4), which are considered to offer a very

accurate description of the flow. However, for a chemically reacting system, the kinetics mechanism models may bring in more uncertainty than fidelity. Besides chemical kinetics models, transport models and coefficients may contain relatively large errors. There is a process that ensures the correctness or appropriateness of the model used in the computation, which is called validation. A validation is usually performed by comparing the obtained numerical solution to suitable experiments, and it checks if the agreement between them is reasonable. In other words, a validation is a process to quantify the modeling error.

The numerical error can be divided into three types, according to Ortega (1990): discretization error, convergence error, and rounding error. The discretization error arises whenever we approximate continuum equations by discrete ones and is the difference between the solutions to the continuum equations and the discrete equations. The convergence error arises when an infinite series is approximated by a finite sum. The rounding error—sometimes called quantization, approximation, or truncation error—is due to the inability of computers to represent real numbers to arbitrary precision. For modern computers, 64-bit floating point description is sufficient to make this error insignificant; 32-bit floating point descriptions may be used in a limited situation with caution.

The discretization error can usually be computed theoretically, while convergence error can be estimated by so-called convergence tests. There usually is no necessity to estimate rounding error, but if desired, one can do so by systematically driving discretization and convergence errors down.

The last kind of error, human error, includes both programmer and user error. Bugs in the coding process, wrong specification of the input parameters, and wrong usage of the code are typical examples of this type of error. This type of error is hard to quantify and is usually the biggest concern of all three. The programming part of the human error, at least the one that affects the accuracy of the numerical results, can be mostly eliminated by going through processes called regression testing (Lakos, 1996) and code verification (Roy, 2005).

Reducing user error is another big challenge in every field of engineering. A good strategy is to use a standardized and easy-to-use human interface. For example, the FORTRAN style inputs, that force users to align inputs exactly as they are read in, is a common source of user errors.

Definition 1 (Verification) *A verification is a process that ensures* $\epsilon_{\text{numerical}}$ *and* ϵ_{human} *are smaller than a tolerance criterion. The verification can be further divided into two categories: code verification that quantifies the programming error and the discretization error, and solution verification that quantifies the convergence error and the user error.*

Definition 2 (Validation) *A validation is a process by which* $\epsilon_{\text{modeling}}$ *is quantified. Validation is possible only when other two kinds of errors, namely* $\epsilon_{\text{numerical}}$ *and* ϵ_{human}*, are known and smaller than* $\epsilon_{\text{modeling}}$*. Therefore, the verification process must be conducted before going through the validation process.*

Models such as turbulence models, transport models, or chemical-kinetics models are ones that must be validated; codes, including an algorithm and its implementation, should be verified but cannot be validated. This fact is important because, quite often, inappropriate verification is used to certify a code. Customarily, numerical data are compared against *similar* experiments to claim that the code is "validated." As will be shown in a later chapter of this thesis, modeling the experimental apparatus without full details is usually not appropriate for validation, let alone the fact that the comparison to experimental data can never provide verification of any kind. Another naive verification often reported is to compare multidimensional simulation results against simpler numerical models such as a one-dimensional model. These may be regarded as a validation of simpler models via multidimensional simulations, but not the opposite.

We will discuss how the current program is implemented and how the object-oriented design can help minimize the probability of bugs and errors in the code, compared to the conventional function-oriented design in the first section.

Another important factor is speed or efficiency. The time it takes to obtain computational results can be broken down to three components: development time, compilation time, and execution time. Although the object-oriented technique was originally adopted to reduce any chance of introducing coding bugs in the code in the present study, it also helped reduce the development and maintenance time significantly as the evolving code became more complex.

There are some drawbacks of coding in C++. One of them is that the resulting code may not be as fast as C or FORTRAN programs in execution time. Another problem is that many, if not all, compilers are not fully compatible with the ANSI C++ standard,

and portability issues arise even when the code is written in a way that conforms to the standard.

3.2 Implementation

3.2.1 The structure of the Omega code

The Omega code developed as part of this project contains approximately 60,000 lines of source code, including header and implementation files, plus external libraries such as MPI (Gropp *et al.*, 1994, 1999) and LAPACK (Anderson *et al.*, 1999; Blackford *et al.*, 1997), which are not counted. Although traditionally computational scientists have preferred the use of FORTRAN programming languages due to performance reasons (Dowd & Severance, 1998), such a function-oriented technique is more error-prone. Object-oriented programming has an edge over function-oriented programming in keeping the bug density in the code small when the techniques are properly used, and therefore, the result is more scientifically reliable.

The implementation of the Omega code follows the protocols for a large-scale programming project, such as described in Lakos (1996), Stroustrup (1991), and Gamma *et al.* (1994).

3.2.1.1 Package structure

First, the static structure of the code is shown in Fig. 3.1. Each package is a collection of source files organized as a physically cohesive unit (Lakos, 1996). For example, the "domain" package is responsible for managing computational domain of the problem, consisting of 19 header files and 16 source files. As can be seen from the diagram, the packages are structured so that the dependency among packages are acyclic and unambiguously levelizable[†], *i.e.*, a level n package may depend on components of level $n-1$ or below only. Such dependency is important not only in implementing codes efficiently but also in maintaining software reliably, as each component can be tested independent of any other packages of the same level or above. These diagrams are described in a language called Unified Modeling Language (UML) and interested readers are encouraged to refer to Miles & Hamilton

[†]Such words as "levelizable" or "levelization" or their meaning may not be found in the Oxford English Dictionary but they are defined and used in Lakos (1996).

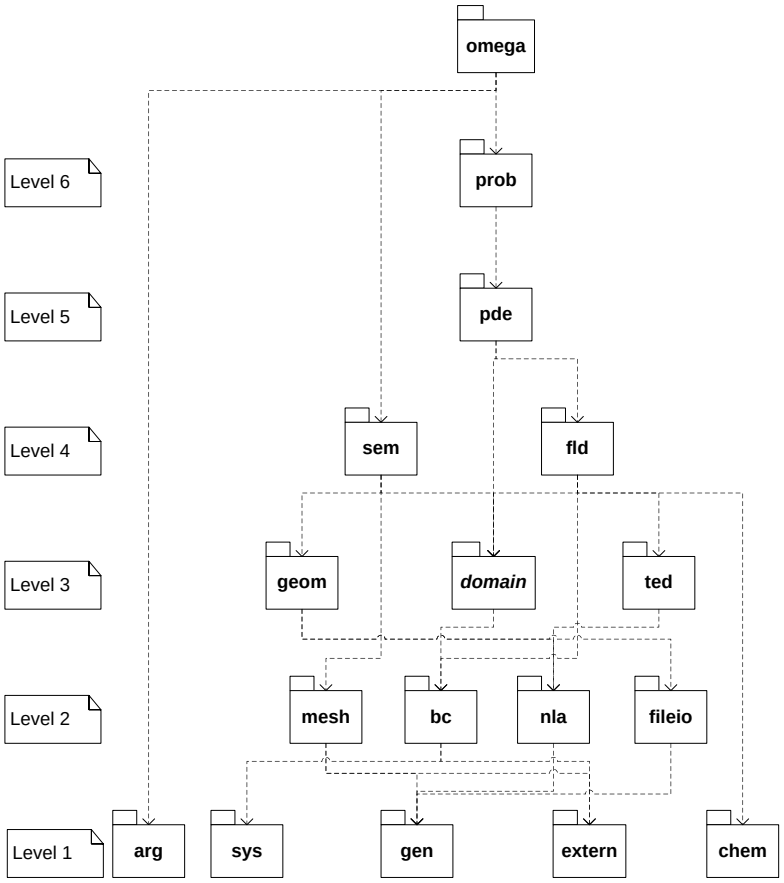


Figure 3.1: A static structure of the packages used in the *Omega* code. Note each package is acyclic and fully levelized, meaning that it can clearly define what other packages it depends on without cyclic dependency. Summary of responsibility of each package is described in Table 3.1.

(2006).

3.2.1.2 Domain structure

One of the problems with so many flow solvers written by so many people is the lack of consistency. Ideally, the whole CFD community should have a single program that computes every flow problem including aerodynamics, supersonic flow (including shockwaves), and low-speed or turbulent combustion. Traditionally, that was not an option because such an approach would make the software less efficient by loading many unnecessary capabilities. However, by not doing so, a significant amount of development time and code verification time is wasted. Although the Omega code does not have such a multipurpose capability currently, it is designed with such extensions in mind. In designing such multi-purpose

Table 3.1: Responsibility of each package

Name	Level	Description
omega		Driver program
prob	6	Problem (*e.g.*, laminar flame) description and specification
pde	5	Partial and ordinary differential equations
sem	4	Spectral element method
fld	4	Field variables such as pressure and velocity
geom	3	Geometry of computational domain
domain	3	Abstract computational domain and coordinate system
ted	3	time-evolving data for multi-step time-integration
mesh	2	Computational mesh within element or domain (1D & 2D)
bc	2	Boundary conditions
nla	2	Numerical linear algebra
fileio	2	Input / output functions
arg	1	Commandline argument processing unit
sys	1	Definition of system dependent variables
gen	1	Generic library (*e.g.*, smart pointers)
extern	1	Wrapper for external library functions
chem	1	Chemistry and transport properties

codes, it is important to extract common structure among different methods and capability and implement them efficiently. Design patterns (Gamma *et al.*, 1994) can be very useful for this purpose.

An example is the bridge pattern used in implementing the computational domain in the Omega code, as in Fig. 3.2. A client who needs to compute the divergence of the velocity field does not need to know in which coordinate system the velocity is defined. All he has to know is that the divergence is defined. How the divergence is computed is passed to the actual coordinate representation such as Cartesian or Cylindrical, for example. Then this coordinate system dictates how to compute the divergence in terms of derivative operators, *i.e.*, $\partial u/\partial x + \partial v/\partial y$ for Cartesian and $\partial u/\partial x + \partial v/\partial y + v/y$ for Cylindrical. These differentiations along coordinates ($\partial/\partial x$ and $\partial/\partial y$) are then evaluated in the actual numerical method implementation, such as the spectral element method (SEM) or finite difference method (FDM). In so doing, one can implement different coordinate systems, dimensions of the problem, and discretization methods in a single code with a uniform structure and interface.

Using this structure, the code can be extended beyond a two-dimensional axisymmetric

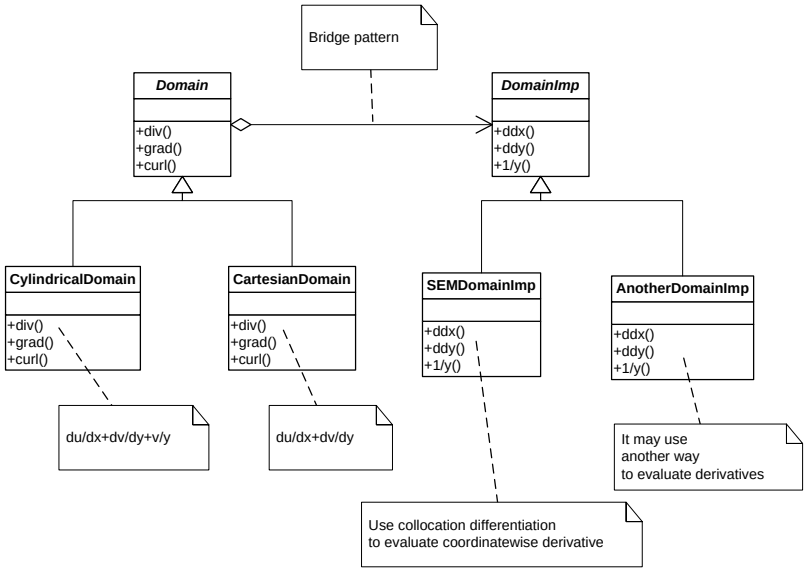


Figure 3.2: This is how the bridge pattern is used in the Omega code. The use of the bridge pattern allows decoupling of the coordinate representation (Cartesian or Cylindrical) from the discretization methods.

problem solver. In fact, the present code has a capability to handle problems in two-dimensional Cartesian coordinates. In addition, it should be easy to implement other discretization techniques, as necessary. One obvious advantage of having a computer program that solves problems in several different coordinate systems is its breadth and the capability to solve different kinds of problems. A less-obvious advantage, but certainly a compelling reason to choose such an implementation, is that it makes verification easy and makes the code more reliable. For example, by sharing many parts of the code when an erroneous outcome is identified that appears only when the code is used to solve problems in a particular coordinate system, then the "bug" is most likely to be in the part that is responsible for that specific coordinate. Since verification can be done in various problems in different coordinate systems, exercising many common components, it gives the user more confidence in the correctness of the code. It also contributes to reducing user errors by maintaining a uniform user interface.

3.2.1.3 Boundary / boundary condition structure

Incorporating boundary conditions is one of the most difficult parts of implementing flow solvers. This difficulty stems in part from the fact that the boundary conditions are as-

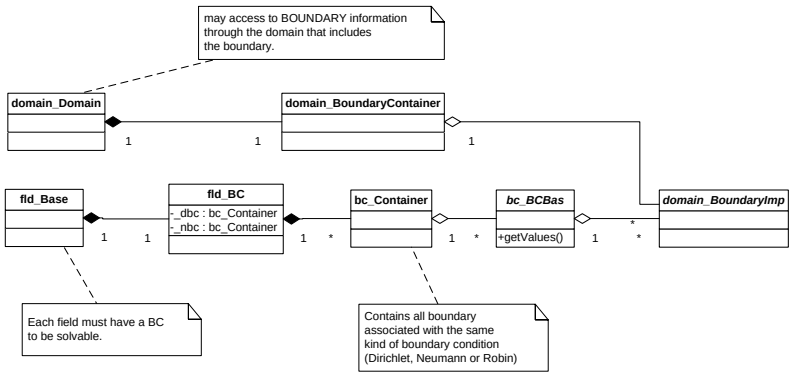


Figure 3.3: Static structure of boundary and boundary conditions

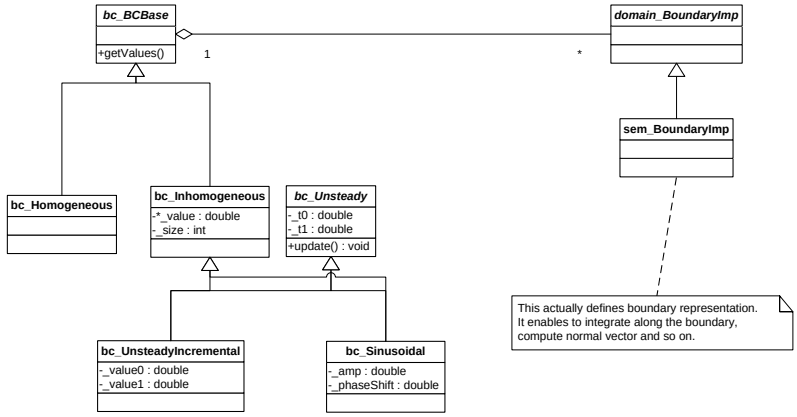


Figure 3.4: Use of bridge pattern in structuring boundary and boundary conditions

sociated with the boundary on which the conditions are defined, as well as conditions by which field values such as velocity or pressure are constrained. To make matters worse, the way in which boundary conditions are applied can depend on discretization methods. This three-way interaction makes the design of the boundary condition implementation a difficult task.

Shown in Fig. 3.3 is the static structure of the boundary and boundary conditions relating to domain and field variables. `fld_Base` is an object that represents a field variable such as velocity, and contains its own boundary conditions, `fld_BC`. The `fld_BC` object then includes two different containers, namely Dirichlet boundary conditions and Neumann boundary conditions. Each container contains as many `bc_BCBase` objects as necessary when that type of boundary condition is used in the problem specification. Each `bc_BCBase` object then contains a pointer or a reference to the boundary object that defines the boundary of the computational domain and numerical value (conditions) associated with this boundary

(and the field). In the other branch, `domain_Domain` is a representation of a computational domain, which contains a `domain_BoundaryContainer` that contains all boundary objects (`domain_BoundaryImp`) which define the boundary of the computational domain. Fig. 3.4 shows the interaction of boundary conditions with their discrete representation. Again the bridge pattern is used to separate the type of boundary conditions—such as homogeneous, inhomogeneous, or unsteady—to a particular representation of the domain—such as the spectral element method.

3.2.2 Procedure in solving matrix equations

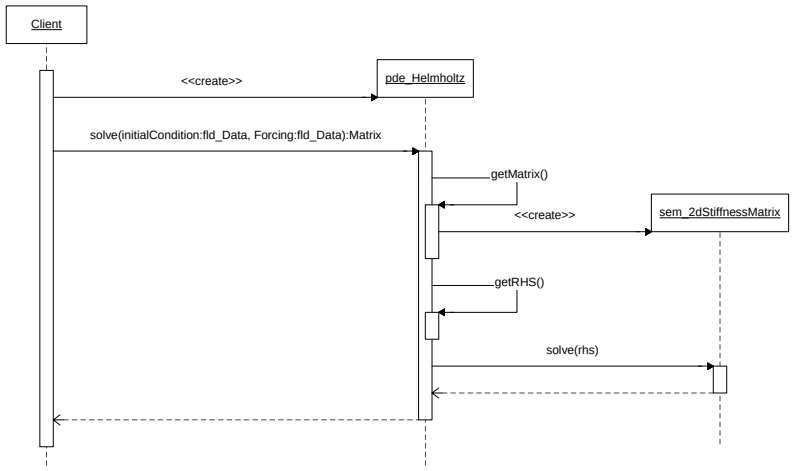


Figure 3.5: The sequence on how the client solves the Helmholtz equation

Shown in Fig. 3.5 is the sequence of how the Helmholtz equations are solved numerically. When a client requests the Helmholtz equation to be solved with a given right-hand side, the `pde_Helmholtz` object forms an appropriate operator matrix as well as the right-hand side vector, and solves the matrix equations. As described in the previous chapter (see page 30), the static condensation and the direct inversion of each resulting matrix is used to solve the linear system. Although an iterative method, such as a conjugate gradient, scales well in a parallel-computing environment, the condition number of the matrix system can be as large as 700 million for one of the cases investigated here, and an iterative method is not a viable option. While preconditioning was not explored in this study, we proceed with the direct method. Fig. 3.6 shows the elaborated sequence of processes used in obtaining the operator matrix for the Helmholtz equation, and Fig. 3.7 shows the sequence of solving the

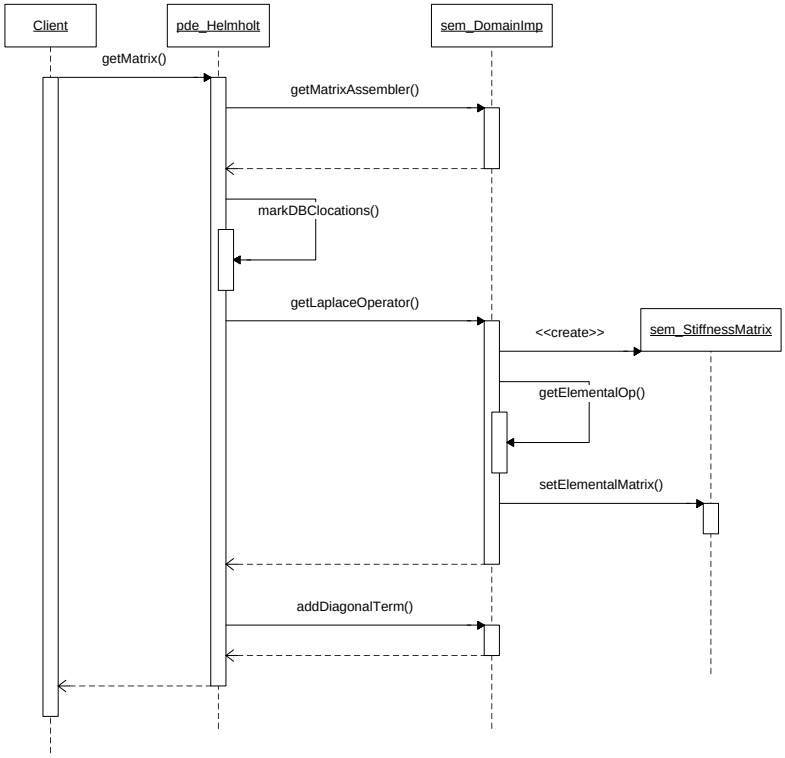


Figure 3.6: When the `getMatrix` method is used, this is the sequence that follows to obtain a weak form of the Laplace operator. Note the `getMatrix` method only computes elemental-level matrices, and assembly of the global operator due to direct stiffness summation is deferred until the assembled matrix is needed.

matrix equation.

3.2.3 Evaluation of transport properties

As discussed in the previous chapter, the viscosity and the self-diffusion coefficients of each species have a dependence on the collision integrals. Since they are a function of temperature only, it is customary to evaluate these transport properties *a priori*, and to tabulate or store them in terms of polynomial coefficients by fitting a polynomial instead of evaluating Eqn.(2.21) or Eqn.(2.31) directly during computation, to save computational time. In the present study, the polynomial fitting approach is adopted following Cantera (Goodwin, 2003), and the viscosity and diffusivities are expressed as follows:

$$\mu_{\mathfrak{m}}(T) = T^{\alpha}\left(\sum_{n=0}^{3}\mu_{\mathfrak{m},n}\left(\log(T)\right)^{n}\right) \tag{3.2}$$

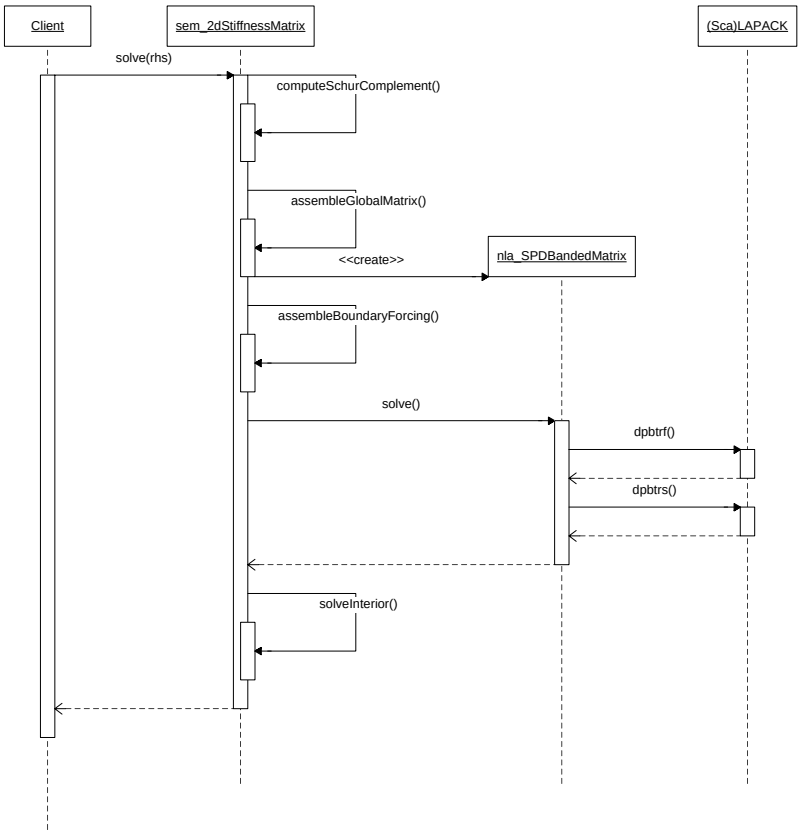


Figure 3.7: When the solve method in `sem_2dStiffnessMatrix` is used, the matrix object tries to factorize the matrix and assemble the global matrix when necessary. For example, `computeSchurComplement` computes A_{11}^* in Eqn.(2.76) at each element level, followed by `assembleGlobalMatrix` that assembles all contributions to A_{11}^* from each element to compute a symmetric positive definite banded matrix. A LAPACK function, `dpbtrf` (or `pdpbtrf` if ScaLAPACK is used), is responsible for Cholesky factorization of A_{11}^*.

$$D_{\mathtt{pq}}(T) = T^{1+\alpha}\left(\sum_{n=0}^{3} D_{\mathtt{pq},n}\left(\log(T)\right)^n\right). \tag{3.3}$$

$\alpha = 0.69$ is used after optimization for the temperature range of 300 K to 2200 K.

3.2.4 Chemistry ODE and Jacobian estimation

The chemical source terms in the species-transport and temperature-transport equations require special treatment because of their stiffness. The temporal evolution of a chemically reacting system is described by chemistry ODEs that arise after splitting the chemical source

terms from the entire governing equations for reacting flows:

$$\frac{dY_{\mathfrak{m}}}{dt} = \dot{\omega}_{\mathfrak{m}} \tag{3.4}$$

$$\frac{dT}{dt} = \dot{\omega}_T = -\frac{\sum_m h_{\mathfrak{m}} \dot{\omega}_{\mathfrak{m}}}{C_{p,\mathrm{mix}}}. \tag{3.5}$$

The right-hand side source terms are evaluated by a Fuego-generated C source code (Aivazis, 2002; Hung, 2003). Fuego is an object-oriented toolkit for chemical kinetics applications developed by Michael Aivazis. It parses a CHEMKIN format chemical mechanism file and produces a C source code that is CHEMKIN link-compatible, but provides increased efficiency by eliminating loops, conditional statements, and other computational overheads. CVODE (Cohen & Hindmarsh, 1994) is used to integrate these equations. Although CVODE can integrate stiff ODEs without a user-supplied Jacobian routine, it is important to have a good estimate of the Jacobian to integrate stiff ODEs such as this one efficiently.

From the Law of Mass Action, we obtain the rate of reaction in molar form,

$$\hat{\omega}_{\mathfrak{m}}(\mathbf{c}, T) = \sum_{k=1}^{K} \hat{\omega}_{\mathfrak{m},k}(\mathbf{c}, T), \tag{3.6}$$

where $\mathbf{c}$ denotes a set of concentrations of each species. This along with a chain rule of derivatives yield the following Jacobian structure in symbolic form:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial \dot{\omega}_{\mathfrak{m}}}{\partial Y_{\mathfrak{n}}} & \vdots & \frac{\partial \dot{\omega}_{\mathfrak{m}}}{\partial T} \\ \cdots & \vdots & \cdots \\ \frac{\partial \dot{\omega}_T}{\partial Y_{\mathfrak{n}}} & \vdots & \frac{\partial \dot{\omega}_T}{\partial T} \end{pmatrix}. \tag{3.7}$$

Let the upper-left submatrix of $\mathbf{J}$ be $\mathbf{J}^0_{mn}$, which is

$$\mathbf{J}^0_{mn} \equiv \frac{\partial \dot{\omega}_{\mathfrak{m}}}{\partial Y_{\mathfrak{n}}} = \mathbf{J}^Y_{mn} + \frac{\bar{W}}{W_{\mathfrak{n}}} \left(\dot{\omega}_{\mathfrak{m}} - \sum_l \mathbf{J}^Y_{ml} Y_{\mathfrak{l}} \right) \tag{3.8}$$

$$\frac{\partial \dot{\omega}_T}{\partial Y_{\mathfrak{n}}} = -\frac{1}{C_{p,\mathrm{mix}}} \left(\sum_m h_{\mathfrak{m}} \mathbf{J}^0_{mn} + \dot{\omega}_T C_{p,\mathfrak{n}} \right) \tag{3.9}$$

where

$$\mathbf{J}^Y_{mn} = \frac{W_{\mathfrak{m}}}{W_{\mathfrak{n}}}\mathbf{J}^c_{mn}, \tag{3.10}$$

and

$$\mathbf{J}^c_{mn} = \frac{\partial \hat{\omega}_{\mathfrak{m}}(\mathbf{c}, T)}{\partial c_n}, \tag{3.11}$$

where m is an index of species $\mathfrak{m}$. Eqn.(3.10) is supplied by the `fejay_` function in the *Fuego*-generated C source code.

Symbolic evaluation of $\partial\dot{\omega}_{\mathfrak{m}}/\partial T$ and $\partial\dot{\omega}_T/\partial T$ terms is more complicated as the explicit form of such derivatives is quite lengthy. We use finite-difference approximations to the last column of the Jacobian matrix.

By using this almost symbolic Jacobian, about a 20% reduction of the number of calls to the function that computes chemical source terms and a 100% improvement in execution speed was achieved in one case.

3.3 Code verification

To make sure that the code does what the developer intends it to do, the importance of verification can never be overemphasized. Within the context of the three errors in numerical simulations mentioned earlier, verification is conducted to quantify errors stemming from numerical errors and the programming part of human error. First, a convergence study of component solvers will be described to make sure that the error decays as we refine spatial and temporal sampling intervals, so that we have a control of discretization errors.

Throughout this study, the error is measured by the L^2 norm,

$$\|f(x,y)\| = \left(\int_\Omega \left(f(x,y)\right)^2 d\Omega\right)^{1/2} \approx \left(\sum_{i,j} w_{i,j} f^2_{i,j}\right)^{1/2}, \tag{3.12}$$

unless otherwise noted.

3.3.1 Verification of components

As described in the previous chapter, the flow solvers for the incompressible Navier-Stokes equations make extensive use of the Poisson equation and the Helmholtz equations. In this section, we will investigate the convergence rate of the Helmholtz equation solver to ensure

that the advertised rate of convergence is achieved.

Fig.(3.8) shows a convergence study of the Helmholtz solver in cylindrical coordinates, for a synthesized problem. The solution is $u(z,r) = \sin(\pi z)\cos(\pi r)$ with even radial parity, and the boundary condition and the forcing term are such that the solution satisfies the Helmholtz equation, $\nabla^2 u + \lambda u = f$ where $\lambda = -\pi^2 \sin(\pi x)\cos(\pi y)$.

In the case of h-refinement, in which polynomial orders are kept constant while the number of elements is increased (thus the element size being refined), it is expected to converge at fifth order; and for p-refinement, in which elements are fixed and polynomial orders are increased, exponential convergence is expected. This property can be seen in Fig.(3.8).

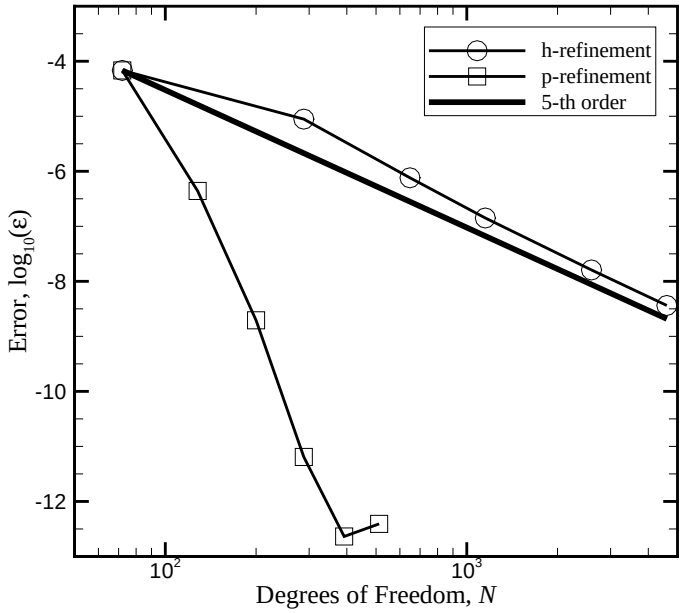


Figure 3.8: h-refinement case shows fifth-order convergence while p-refinement shows faster convergence.

3.3.2 Verification by the method of exact solution

One of the advantages of being able to solve Cartesian problems in addition to cylindrical ones within the same software is that it makes it possible to conduct verification tests that are otherwise not possible. Kovasznay flow in a Cartesian coordinate system is one of the

Table 3.2: Convergence data of Helmholtz equation solver

K	Q	N	N_{Net}	Error
2	6	72	32+34	6.81278e-005
8	6	288	128+103	8.87486e-006
18	6	648	288+208	7.61115e-007
32	6	1152	512+349	1.41611e-007
72	6	2592	1152+739	1.61071e-008
128	6	4608	2048+1273	3.64283e-009
2	8	128	72+48	4.39157e-007
2	10	200	128+62	1.96032e-009
2	12	288	200+76	6.43544e-012
2	14	392	288+90	2.30857e-013
2	16	512	392+104	3.9094e-013

few exact solutions of the Navier-Stokes equations that exercise every term in the equation (Kovasznay, 1948).

The Kovasznay flow is described by

$$u(x, y; \lambda) = 1 - \exp(\lambda x)\cos(2\pi y) \tag{3.13}$$

$$v(x, y; \lambda) = \frac{\lambda}{2\pi}\exp(\lambda x)\sin(2\pi y) \tag{3.14}$$

$$p(x; \lambda) = 1 - \exp(2\lambda x)/2 \tag{3.15}$$

where

$$\lambda = Re/2 - \sqrt{Re^2/4 + 4\pi^2}. \tag{3.16}$$

This solution has been used in verification in the past (Karniadakis *et al.*, 1991; Blackburn & Sherwin, 2004). In particular, Blackburn & Sherwin (2004) used this exact solution to verify their implementation in three-dimensional cylindrical coordinates by translating this solution in two-dimensional Cartesian coordinates into a three-dimensional cylindrical coordinate system.

As shown in Fig. 3.9, the expected rate of convergence is observed using h-refinement with seventh-order accuracy, as well as the high-order convergence using p-refinement.

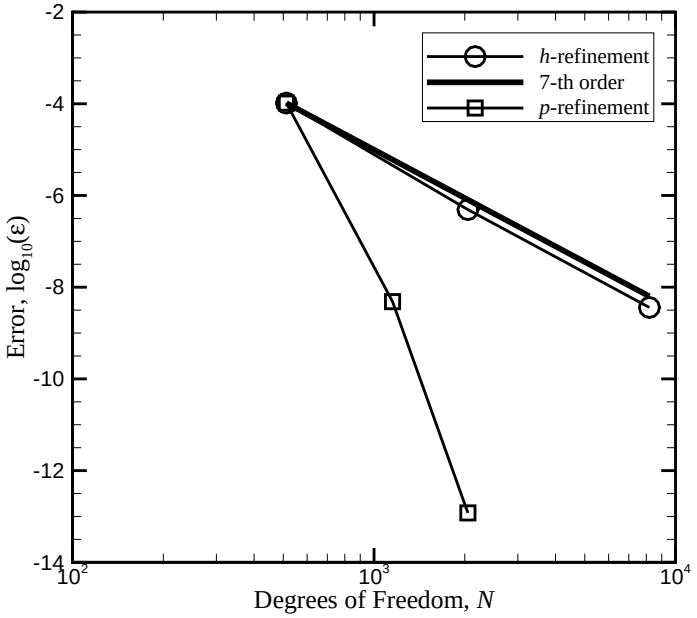


Figure 3.9: h-refinement case shows seventh-order convergence while p-refinement case shows faster convergence.

3.3.3 Verification of the code via the method of manufactured solution

The verification of a code through an exact solution is very powerful. However, quite often there is no exact solution available in closed form without oversimplifications. The method of manufactured solution is universally applicable to verification of a code whether or not an exact solution is available (Roy, 2005). In the present study, an exact solution based on the Kovasznay flow solution is used to create a problem with an exact solution:

$$\rho(z,r) = \exp(\lambda z/2) \tag{3.17a}$$

$$u(z,r) = \exp(-\lambda z/2) - \exp(\lambda z/2)\cos(2\pi r) \tag{3.17b}$$

$$v(z,r) = \lambda \exp(\lambda z/2)\sin(2\pi r)/(2\pi) \tag{3.17c}$$

$$p(z) = (1 - \exp(2az))/2 \tag{3.17d}$$

$$T(z,r) = T_0/(\exp(C_0 z)(C_1 + C_2\cos(2\pi r) + C_3\sin(2\pi z))) \tag{3.17e}$$

$$Y_{\mathrm{H2}} = Y_{10} + Y_{11}\sin(2\pi z) + Y_{12}\cos(2\pi r) \tag{3.17f}$$

$$Y_{\mathrm{N2}} = 1 - Y_{10} - Y_{11}\sin(2\pi z) - Y_{12}\cos(2\pi r) \tag{3.17g}$$

where $T_0 = -688.0862031843639$, $C_0 = (5 - \sqrt{25 + 4\pi^2})/2$, $C_1 = -9.815178$, $C_2 = C_3 = -1.299873$, $Y_{10} = 0.3$, $Y_{11} = Y_{12} = 0.05$ are used.

This solution satisfies the divergence constraint, $\nabla \cdot \rho\mathbf{u} = 0$, and the mass-fraction constraint, $\sum Y_{\mathfrak{m}} = 1$. The exact solution of temperature is somewhat complicated due to its coupling to density and mass fraction fields through the equation of state. The required forcing term to balance these manufactured solution on the governing equations is computed using *Mathematica*®, and the C source code generated by *Mathematica*®is used to evaluate the forcing terms.

Figure 3.10 shows the decay of errors for both h-refinement, in which the number of elements is increased with fixed polynomial order, and p-refinement, in which polynomial order is increased with the number of elements fixed.

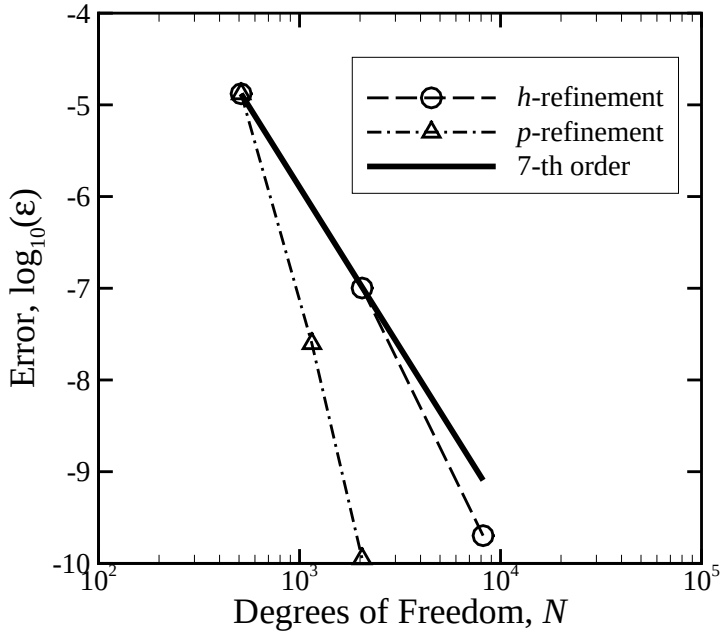


Figure 3.10: The method of manufactured solution provides verification of the code. h-refinement case shows seventh-order convergence while p-refinement case shows faster convergence. Both error decay rates are in line with the expected rate.

3.3.4 Verification of the code against another numerical solution

Although this test is not exactly verification in the sense of the definition given above, it is still a useful comparison to gain confidence on the newly developed code. Frouzakis *et al.*

(1998) reported non-premixed opposed-jet hydrogen flames at a Reynolds number of 100, using a spectral element method and the hydrogen oxidation kinetic mechanism by Yetter *et al.* (1991) (H4 mechanism in Table 1.1). The opposed jet flame is very similar in nature to a stagnation flame, and therefore, is an appropriate test case for the present work. We have set up a computational domain using the same elements, as reported in Frouzakis *et al.* (1998), with the same polynomial order. Shown in Fig. 3.11 is the computational domain used for this study. Hydrogen diluted with nitrogen is introduced from the right between $0 \leq r \leq 0.005\,\mathrm{m}$ with the inlet velocity of $0.15925\,\mathrm{m/s}$, and air is introduced from the left at the same velocity. The top part of the domain ($r = 0.02\,\mathrm{m}$) is an outflow boundary condition, and the rest is an isothermal wall, kept at $300\,\mathrm{K}$.

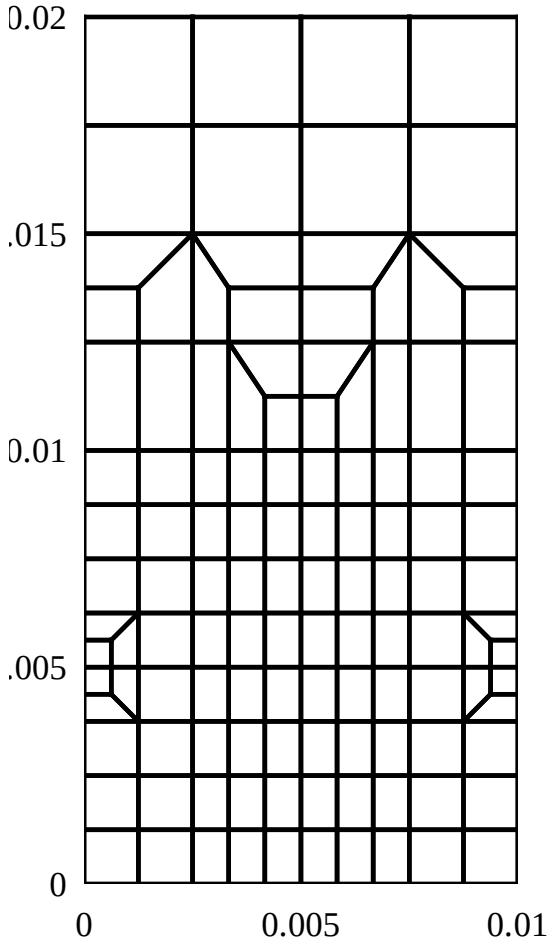


Figure 3.11: Computational domain for opposed-jet hydrogen flame study taken from Frouzakis *et al.* (1998). It includes 120 elements; each contains 6 by 6 collocation points.

Shown in Figs. 3.12–3.15 are comparisons of hydrogen and hydroxyl radicals, temperature, and axial velocity profiles between the Omega code, developed in the present work, and the two-dimensional numerical solution by Frouzakis *et al.* (1998). One-dimensional solutions using CHEMKIN, reported in Frouzakis *et al.* (1998) are also plotted in each figure for reference. The same hydrogen kinetics model (Yetter *et al.*, 1991) is used in all cases. Despite some differences in the formulation and transport models, the two two-dimensional solutions are in good agreemeent.

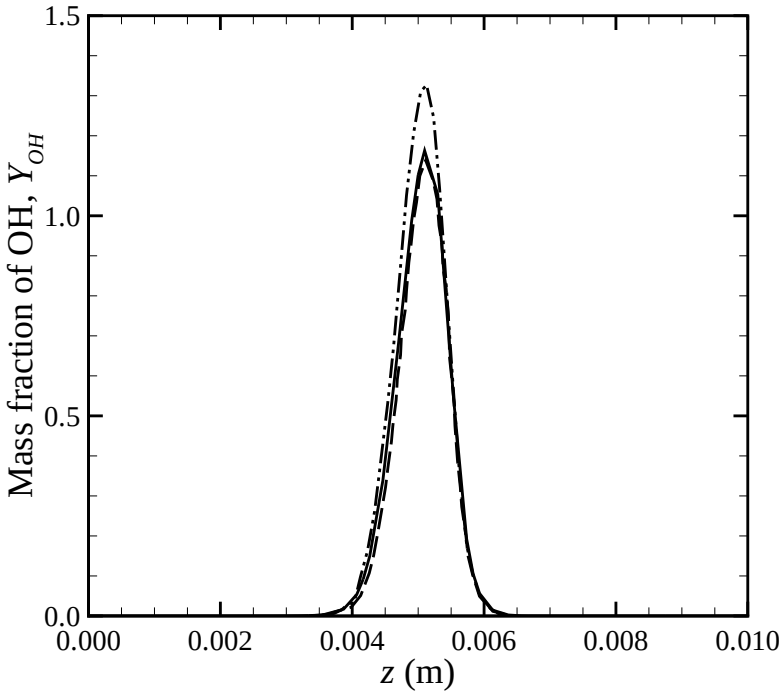


Figure 3.12: The mass fraction of the hydroxyl radical profile is compared among the Omega code (solid line) developed in the present work, the two-dimensional solution reported in Frouzakis *et al.* (1998) (dashed line), and the one-dimensional solution using CHEMKIN also reported in Frouzakis *et al.* (1998) (double-dot-dashed line).

3.4 The validation of the model

Validation refers to comparison to experimental data or a theoretical prediction that can support the correctness of the numerical solution. For example, it is easy to verify that the numerical solution actually satisfies the set of equations by looking at the residuals after substituting the numerical solutions to the equations. However, this does not mean that the numerical solution is the right one; there might be a modeling error. In addition to

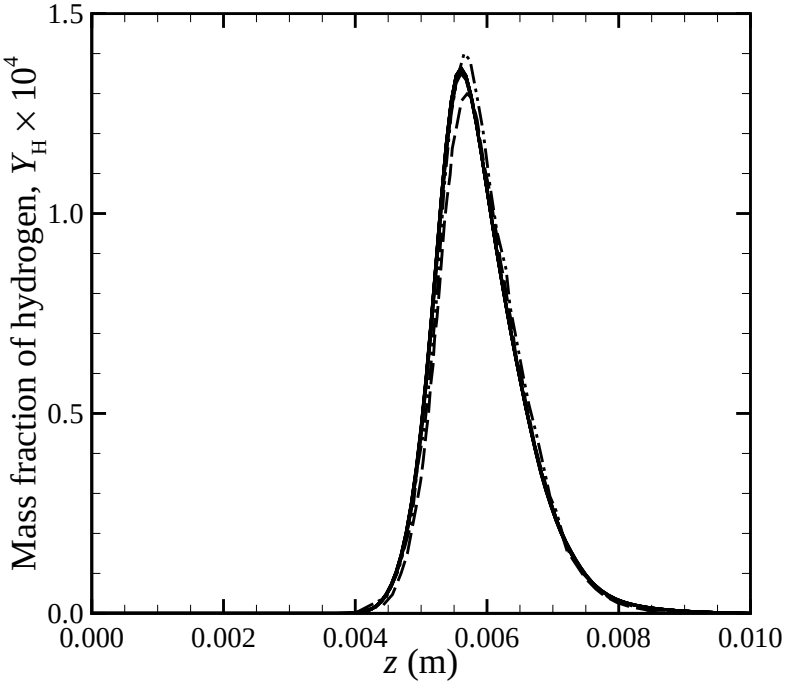


Figure 3.13: Comparison of the mass fraction of hydrogen atom profile. (Legend as in Fig. 3.12.

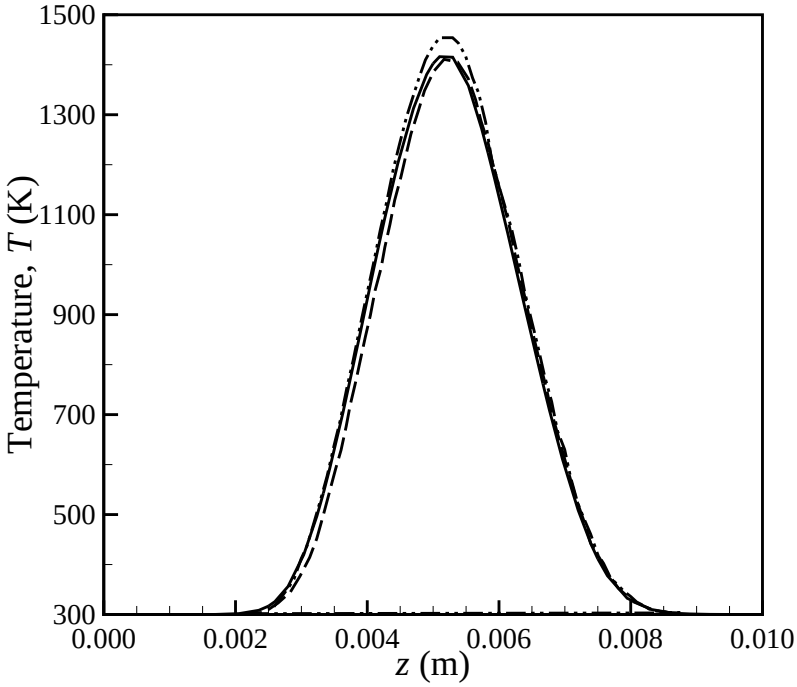


Figure 3.14: Comparison of the temperature profile. (Legend as in Fig. 3.12.)

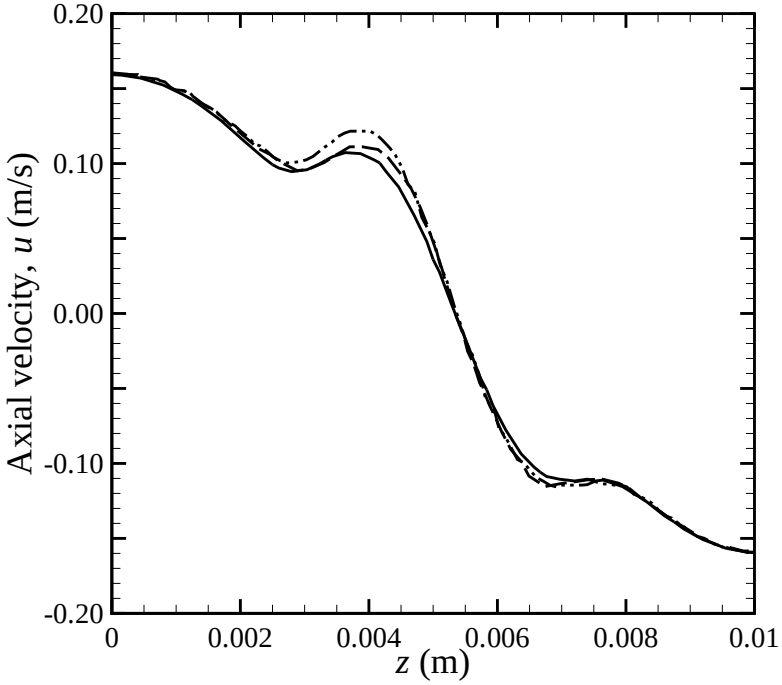


Figure 3.15: Comparison of the axial velocity profile. (Legend as in Fig. 3.12.)

modeling of fluid mechanics, there are three model components used in the numerical study of reacting flows: thermodynamics (such as enthalpy and heat of formation), the transport model and coefficients, and the chemical kinetics. Although each of the three components needs to be improved, and is improving today, it is expected that the chemical kinetics model contains more uncertainty and error (Williams, 2000). The validation of reacting-flow models is one of the main themes of the present study and will be discussed in the following chapter.

It is worth noting that the modeling error, $\epsilon_{\text{modeling}}$ in Eqn. (3.1), can be quantified only by simulating experiments directly. For example, if the study of an ideal stagnation flame that extendes to infinity in radial direction with perfectly flat flame shape is considered, and this ideal setting is modeled in both experiments and numerical study, it is impossible to quantify $\epsilon_{\text{modeling}}$ because experiments represent a different situation since such an ideal flame cannot be realized in the laboratory. However, by simulating flames observed in a laboratory directly, instead of modeling idealized flames, it is possible to quantify the modeling error that appears in mathematical models, $\epsilon_{\text{modeling}}$. The availability of the recent high-quality experimental data and all necessary experimental implementation details from

the work of Bergthorson *et al.* (2005b) and Bergthorson (2005), along with the necessary computational resources and support, permitted this approach to be adopted in the present study.

Chapter 4

Stagnation flow and flame simulations

Knowledge itself is power.—Francis Bacon

4.1 The laminar impinging jet—Cold flow

4.1.1 Introduction

Axisymmetric jets impinging perpendicularly on a wall are encountered in a variety of real-life situations such as Chemical Vapor Deposition (CVD) processes (Goodwin, 2003). Although the fluid mechanics of the wall-impinging jets are interesting phenomena in their own right, this topic is studied here for two reasons. The first is to assess the extent to which experimental data and numerical data agree when no chemical reactions are involved. In particular, since this is the case in which the modeling error is expected to be small as the incompressible Navier-Stokes equations are known to describe uniform density flow well, any uncertainty between measurements and computation will be quantified. The second objective is to learn the general behavior of stagnation flow to apply the knowledge of non-reacting flow to reacting cases. In this study, the nozzle-to-plate separation distance L to nozzle-diameter d ratio range of $0.5 \leq L/d \leq 1.5$ is studied. This range of L/d has been investigated experimentally and is useful in the study of strain-stabilized flames in combustion research.

This type of flow has a well-known approximate similarity solution presented in Schlichting (1960), called Hiemenz flow. This similarity solution is based on an analytical solution for potential flow and used to develop the one-dimensional stagnation flame or opposed-jet

flame model by Kee *et al.* (1988). This was a very useful extension in combustion research and many studies use this one-dimensional formulation, although it has rarely been used to study non-reacting flow, as more-detailed multidimensional simulations are available for such cases. However, there are not many numerical studies reported for the separation distances that enable us to draw a conclusion about whether a one-dimensional model is a good approximation for stagnation flows with small separation distances. For example, Deshpande & Vaishnav (1982) studied an axisymmetric round jet impinging on a wall for $L/d = 1.5$ and 2. Pelletier *et al.* (2004) also studied axisymmetric laminar and turbulent impinging round jets using the finite element method, at separation distances of $L/d = 1$, 2, and 4. Nothing shorter than these separation distances has been reported to the author's knowledge.

In the present study, a simulation of cold flow laboratory experiments reported by Bergthorson *et al.* (2005a) is conducted by including parts of the nozzle-plenum assembly in the computational domain as well as the stagnation plate. A smaller flow enclosure is used compared to the original experiments to save computational time without sacrificing accuracy. In addition, the behavior of the one-dimensional model developed by Kee *et al.* (1988) is assessed.

For the axisymmetric two-dimensional simulation of a laminar impinging jet with a separation distance $L/d = 1.424$, the computational domain shown in Fig. 4.1 is used, which appropriately models the corresponding experiments conducted by Bergthorson *et al.* (2005a).

A study of the effects of boundary conditions on the flow field was undertaken to ascertain that the flow field near the stagnation point was insensitive to the particular choice of the boundary conditions away from the region of interest. A nearly flat velocity profile is introduced at the nozzle inlet, just downstream of the last turbulence-management grid and before the beginning of the nozzle contraction (Fig. 4.1): $u_1(r)/U_\mathrm{P} = \tanh[c_1(1 - r/r_\mathrm{P})]$, where $r_\mathrm{P} = d_\mathrm{P}/2$ is the radius of the plenum, U_P is the centerline velocity at the plenum, and c_1 is set to 50. This profile mimics the outflow from the turbulence-management section in the experiments. As expected, the jet profile at the nozzle exit is insensitive to the choice of inlet profile, owing to the high contraction ratio in the nozzle design. Shown in Fig. 4.2 is a simulated axial-velocity profile at the nozzle exit compared with a Pitot-probe measurement along a line that cut through the axis. Both simulation and experiments are conducted

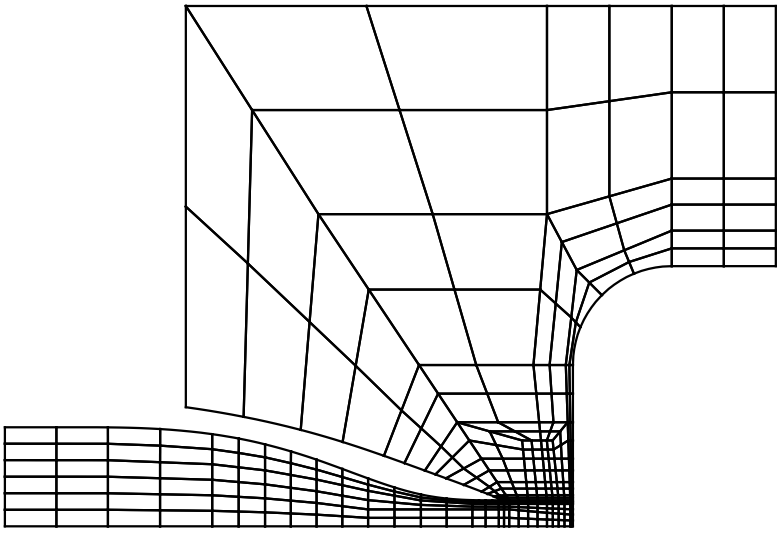


Figure 4.1: A computational domain used for the simulation of non-reacting stagnation flow. It incorporates 360 elements, within which 6 by 6 mesh collocation points are used.

without the stagnation plate to observe the characteristics of the nozzle. The well-designed contraction nozzle produces a flat velocity profile, and the simulation, which incorporates the shape of the nozzle interior as specified in Bergthorson *et al.* (2005a), captures the velocity profile accurately. The slight disagreement between simulation and experiment in the wall boundary layer region is attributable to the finite pitot-probe extent, d_{pitot}, in the radial direction, for which no corrections were applied (Bergthorson *et al.*, 2005a).

To simulate the entrained flow, an entrainment flux is introduced through the lower portion of the outer domain (top left). It was shown that the particular choice of entrainment flux does not affect the velocity field in the near-field region of interest ($0 \leq r/d \leq 1$, $-L/d \leq z/d \leq 0$), more than a percent (Bergthorson *et al.*, 2005a). A uniform-pressure condition is specified at the outflow between the flow enclosure and the stagnation plate (top right).

Results presented in this section were presented in Bergthorson *et al.* (2005a).

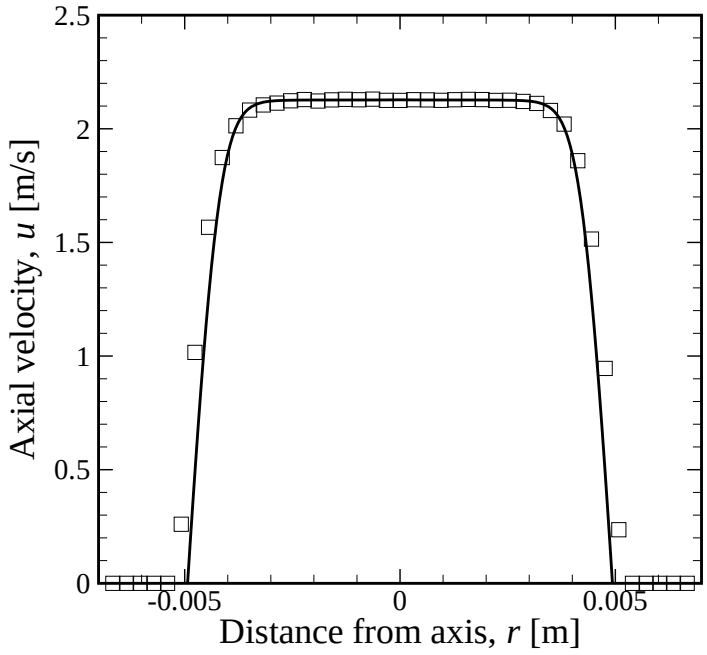


Figure 4.2: Axial velocity profile at the nozzle exit. The internal diameter of the nozzle at the opening is 9.9×10^{-3} m and the Reynolds number of the jet based on the centerline axial velocity is about 1400. (Solid line) simulation; (□) measurements

4.1.2 Results and discussion

4.1.2.1 Comparison to experiments

Experimental velocity data reported in Bergthorson *et al.* (2005a) were recorded at three nominal Reynolds numbers,

$$Re \equiv \frac{\rho\, d\, u_{\mathrm{B}}}{\mu} \cong 400\,,\ 700\,,\ \text{and } 1400, \tag{4.1}$$

with actual values within ±35 in each case, and at three nozzle-to-stagnation-plate separation distance to nozzle-exit-diameter ratios, $L/d \cong 0.7$, 1.0, and 1.4. The reference velocity, u_{B}, is called the Bernoulli velocity and is defined by

$$u_{\mathrm{B}} = \sqrt{\frac{2\Delta p/\rho}{1-(d/d_p)^4}}. \tag{4.2}$$

One could use the centerline axial velocity at the nozzle exit, u_0, as the reference velocity. This is not appropriate because as the wall-nozzle separation distance decreases, the adverse

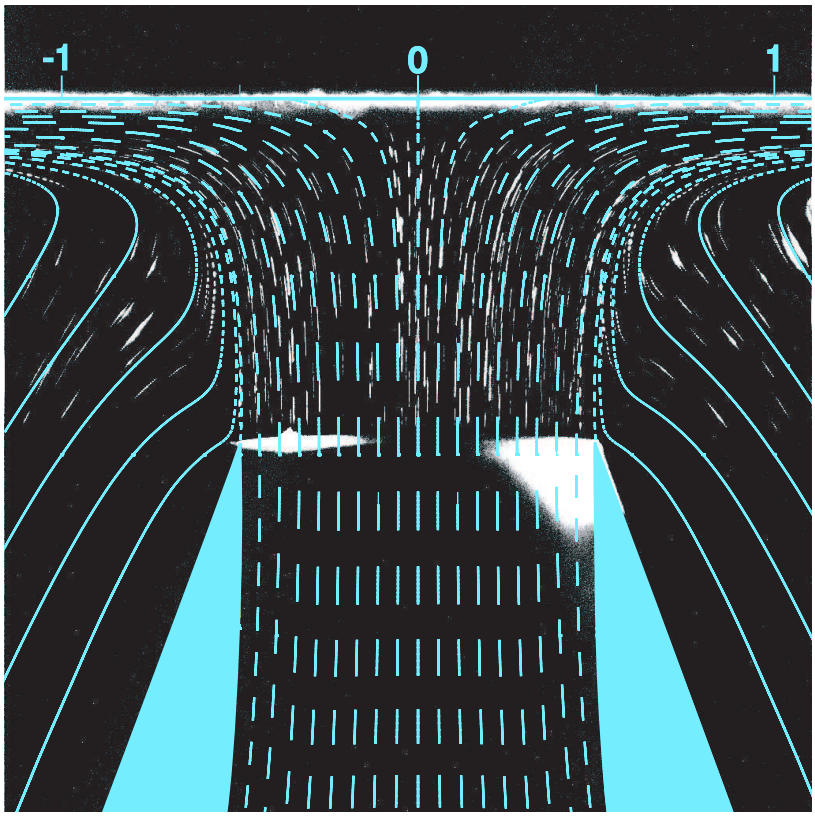


Figure 4.3: Particle streak image (monochrome) detailing entrained flow with superimposed axisymmetric viscous calculation (blue lines) at $Re = 700$ and $L/d = 1.0$

pressure gradient created by the stagnation point propagates upstream, inside the nozzle, and the exit axial velocity tends to have a larger velocity deficit. We term this the "wall-nozzle proximity effect" in this study. For large separation distances of $L/d > 1$, $u_0 \sim u_B$, however for smaller cases, say $L/d < 1$, as often employed to approximate idealized stagnation flow, u_0 is measurably smaller than u_B.

Fig. 4.3 compares particle-streak-image data and streamlines from the axisymmetric viscous simulations along the centerline. Good qualitative agreement can be seen, even in the entrainment region where the velocities are low ($< 0.02\, u_B$).

Fig. 4.4 shows axisymmetric simulation results on top of experimental data from (Bergthorson *et al.*, 2005a) at Re =400, 700, and 1400, respectively, for $L/d = 1.4$, which is the largest separation distance studied in Bergthorson *et al.* (2005a). All velocity data are taken along the axis and normalized by the Bernoulli velocity, u_B.

Although the code has gone through extensive verification processes, this cold flow simulation provides another layer of testing as the incompressible Navier-Stokes equa-

tions are regarded as a high-fidelity model of uniform-density flow. Within the range of Reynolds numbers studied, simulation and experiments are in good agreement. In many cases, the root-mean-square error of experimental data against numerical solution, which is a piecewise-smooth polynomial, is less than one percent of the reference velocity. Even the largest error was within 1.5%. Provided computed data contain rounding errors only, the small scatter in the experimental data are useful for the validation of chemical kinetics models that will be discussed in the next section.

When the ratio of the wall-to-nozzle separation distance to the nozzle diameter, L/d, is varied from 1.4 to 0.5, axial velocity profiles fall on top of a single curve when scaled by Bernoulli velocity. This is consistent with the finding reported in Bergthorson *et al.* (2005a). Fig. 4.5 gives pressure contours at $L/d = 0.5$ and 1.4, with pressures scaled by the Bernoulli pressure, which is the pressure drop across the nozzle. The near-wall pressure field is not significantly altered by changes in the nozzle position, and this fact explains the Bernoulli velocity scaling.

4.1.2.2 Assessment of the one-dimensional model

Although the one-dimensional model by Kee *et al.* (1988) was designed for axisymmetric stagnation flames, it also applies to nonreacting flow. In this case, the only equation to solve is the radial momentum equation, Eqn.(2.7), which is a third-order nonlinear ordinary differential equation. The one-dimensional model is based on a constant-pressure-eigenvalue assumption. The numerical evidence of the variation of the pressure eigenvalue is shown in Fig. 4.6. Although the variation in the radial direction is not significant, the profile shows nonuniformity in the axial direction. The profile is close to zero around the nozzle exit, if there is enough separation between the nozzle lip and the wall, but has an observable variation in the stagnating-jet region.

Fig. 4.7 compares the two-dimensional simulation velocity data, at $Re = 700$, to four different one-dimensional simulations, with plug-flow boundary conditions and different choices of the interior boundary location, ℓ. Solutions with plug-flow boundary conditions do not agree with the two-dimensional simulation data at any ℓ/d. When $\ell/d \geq 1.0$, the plug-flow (no velocity gradient) assumption provides a good approximation, as is evident from the figure where the very begining of the one-dimensional results agree with the two-dimensional data. However, one-dimensional solutions soon underpredict the axial velocity.

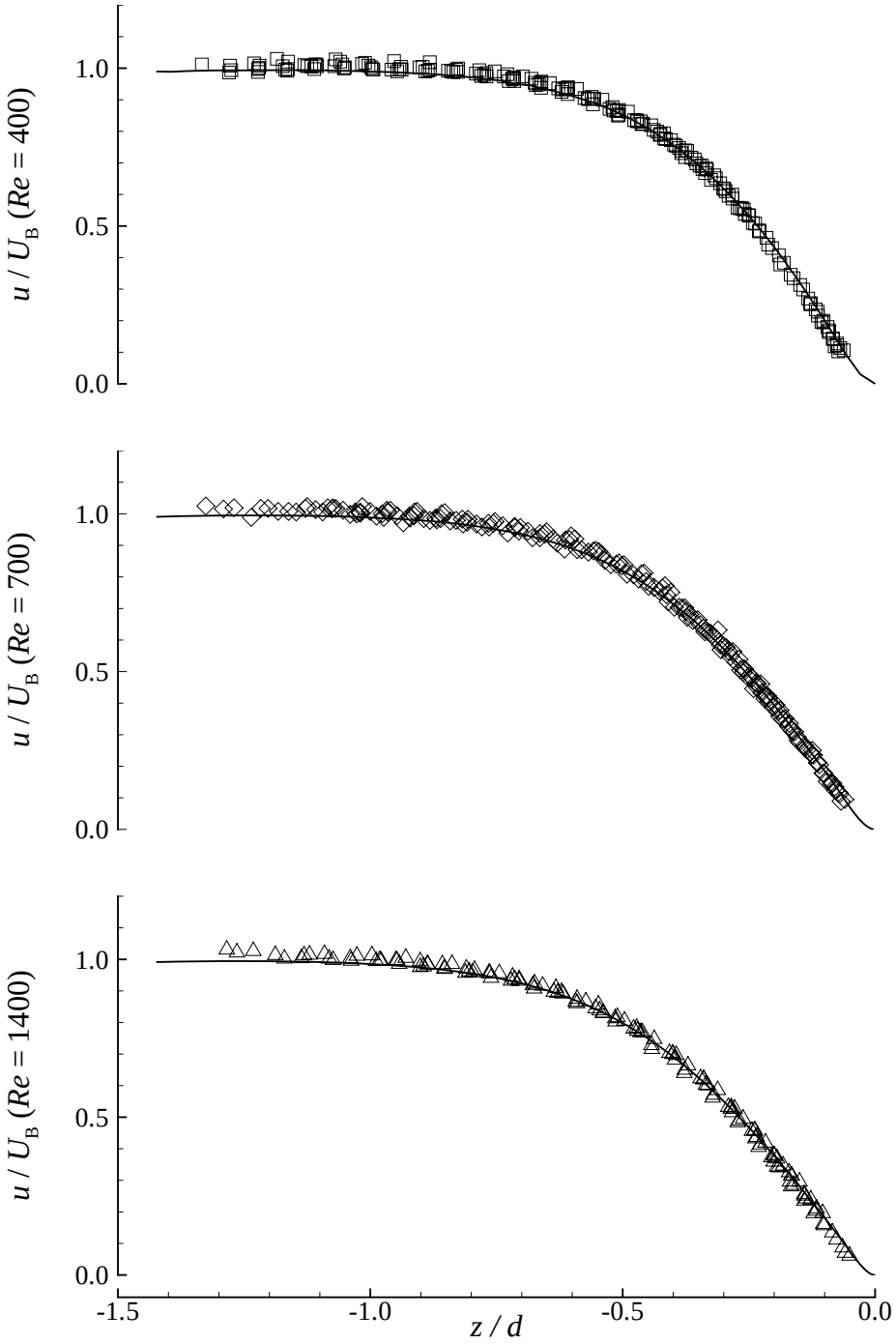


Figure 4.4: Axial velocity profiles along the axis of symmetry for $Re = 400$ (top), $Re = 700$ (middle), and $Re = 1400$ (bottom) with nozzle lip located at $z/d = -1.414$. Solid lines indicate two-dimensional simulations, while symbols indicate laboratory measurements by J. Bergthorson.

On the other hand, when $\ell/d \leq 0.8$, finite velocity gradients are evident and the plug-flow boundary condition is incorrect. In this case, the one-dimensional model tends to

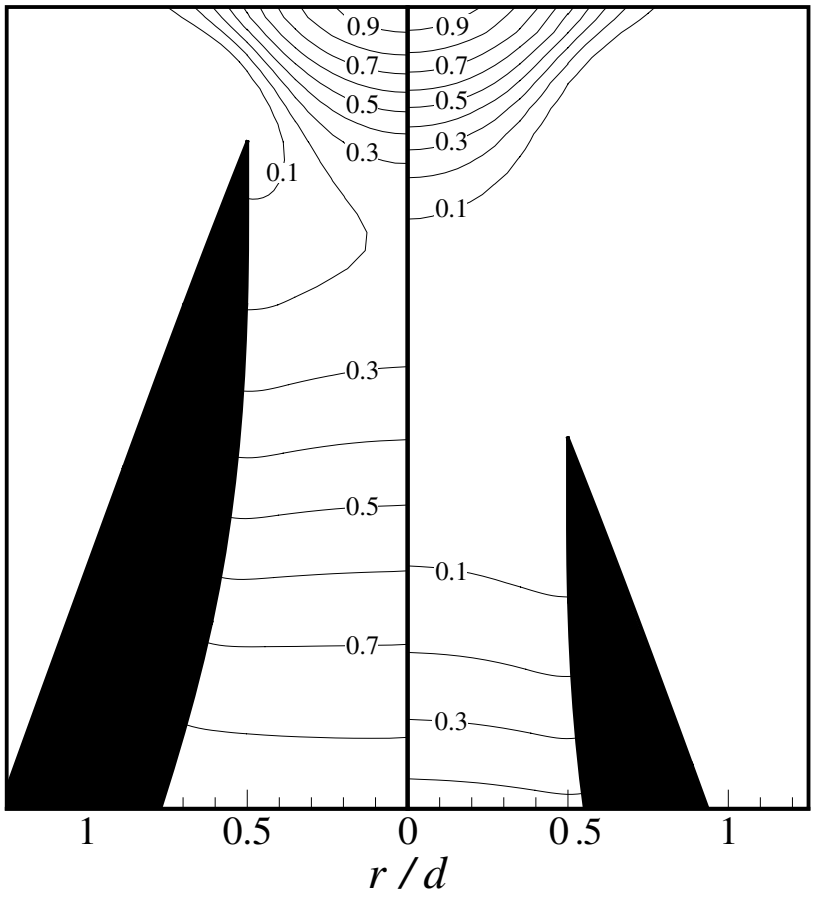


Figure 4.5: Pressure isocontours normalized by Bernoulli pressure at $L/d = 0.5$ (left) and $L/d = 1.4$ (right). (From Bergthorson *et al.*, 2005a)

overpredict soon after from the left boundary to the right boundary.

This is because the outer solution to the one-dimensional cold-flow equations is a parabola, as pointed by Davis *et al.* (2002b), which is a consequence of the uniform-eigenvalue assumption, and cannot capture the free-jet behavior (zero-gradient region of flow) exhibited for $z/d > 1.0$. However, this is the only questionable assumption in the one-dimensional model, and if $\Lambda(z)$ could be modeled or provided to the model by any means, the one-dimensional model should be able to predict flow fields along the axis well.

Since the plug-flow boundary condition works for no single case, a different boundary condition must be used for a one-dimensional model to be useful. The velocity and velocity-gradient boundary conditions at a given axial location, $U(\ell)$ and $U'(\ell)$, can be specified from a fitting function to the two-dimensional simulation data or the experimental data [see Eq. (4.3)], as described in Bergthorson *et al.* (2005a) and briefly discussed below.

The one-dimensional solution calculated using this method at $Re = 700$, over the range $0.3 \leq \ell/d \leq 0.7$, has a maximum error of less than 3% of u_{B} when compared to two-dimensional simulations.

4.1.2.3 Scaling parameters and error-function axial velocity profile model

As expected, for the flows investigated, Reynolds number is found to be a relatively weak parameter. Fig. 4.8 shows the maximum centerline axial velocity-gradient decreasing with increasing Reynolds number when scaled in this manner. The main effect of Reynolds number is the change in the effective jet diameter as a consequence of the boundary-layer displacement thickness. When u/u_{B} is plotted versus z/d_*, where d_* is the diameter corrected for nozzle-wall displacement thickness, the data collapse for all Reynolds numbers (Fig. 4.9) indicating minimal viscous effects in the free flow in the range of Reynolds numbers investigated. It is interesting to point out that this discovery was inspired by results of simulations without the nozzle-plenum assembly, *i.e.*, the inflow boundary conditions specified at the nozzle exit.

Kostiuk *et al.* (1993), in their study of turbulent jets, showed that opposed- or impinging-jet velocity data are well-characterized by an error function and used the parameters obtained from the error-function fit to collapse their experimental data. Their error function contained three adjustable parameters, the velocity at infinity U_∞, a strain-rate parameter α, and a wall-offset length δ/d,

$$\frac{u(z)}{U_\infty} \;=\; \mathrm{erf}\left[\,\alpha\,\left(\frac{z}{d} \,-\, \frac{\delta}{d}\right)\right] \;. \tag{4.3}$$

The collapse of the experimental and numerical data discussed above suggests that the appropriate velocity scale for laminar impinging jets is the Bernoulli velocity, *i.e.*, $U_\infty = u_{\mathrm{B}}$ as described in Bergthorson (2005) and Bergthorson *et al.* (2005a). From one-dimensional viscous stagnation-flow theory (see Bergthorson *et al.*, 2005a), the scaled-offset length, δ/d, which is proportional to the scaled wall boundary-layer thickness, can be related to the strain-rate parameter α, such that,

$$\frac{\delta}{d}(Re, \alpha) \;=\; 0.755\sqrt{\frac{1}{Re\,\alpha}} \;. \tag{4.4}$$

Thus, the only free parameter in this error-function fit to the data is the strain-rate parameter α, which should be a function of Reynolds number alone, *i.e.*, $\alpha = \alpha(Re)$. Therefore, the axial velocity field for an axisymmetric impinging laminar jet is fully specified by the Bernoulli velocity u_{B}, since the Reynolds number is, in turn, derived from it (Bergthorson *et al.*, 2005a).

The error-function fit to the numerical data at $Re = 1400$ is plotted in Fig. 4.10. The error function was fit to each experimental and simulation case by adjusting α such that the root-mean-squared (rms) error was minimized. For each Reynolds number, the strain-rate parameter α was averaged over the range $0.7 \leq L/d \leq 1.4$. This single $\alpha(Re)$ dependence was subsequently used in all error-function fits to determine the resulting rms error ϵ_{rms}. The fit parameters and ϵ_{rms} are shown in Table 4.1.

Table 4.1: Error-function fit parameters and rms error ϵ_{rms} of fits to experimental and viscous-simulation data.

			Experiment	Simulation
Re	α	δ/d	$\epsilon_{\mathrm{rms}}/u_{\mathrm{B}}$	$\epsilon_{\mathrm{rms}}/u_{\mathrm{B}}$
400	2.21	0.027	0.017	0.014
700	2.00	0.020	0.010	0.009
1400	1.88	0.015	0.011	0.010

This is very useful for determining suitable boundary conditions for one-dimensional simulations. Specifying the Bernoulli velocity, the Reynolds number is known and the strain-rate parameter, α, can be determined from Table 4.1. δ/d can be calculated from Eqn.(4.4). The analytical expression given by Eqn.(4.3) can be used to calculate the velocity profile along the axis.

4.1.2.4 Effect of nozzle design for stagnation-flow experiments

As previously mentioned, the main Reynolds number effect for this flow is through the nozzle-wall boundary-layer thickness. The effect of the nozzle-exit velocity profile is studied in Fig. 4.11 for profiles varying from a top-hat shape, representative of the outflow from a high-contraction ratio nozzle, to a parabolic profile, representative of fully developed laminar pipe flow. Real nozzle-exit velocity profiles should lie between these two extremes. Intermediate cases are studied by specifying hyperbolic tangent profiles whose coefficients are adjusted to represent the variation of boundary-layer displacement-thicknesses. The results in Figs. 4.11 and 4.12 are obtained by removing the nozzle interior from the axisym-

metric two-dimensional simulation domain and specifying the velocity profiles at the nozzle exit. Due to the lack of a plenum in the simulations, velocities are scaled by the velocity at the axis of the jet, u_0, instead of the Bernoulli velocity. Fig. 4.11 indicates that there is a significant effect of the nozzle-exit velocity profile on the resultant axial velocity field. Fig. 4.12 plots the axial velocity profiles with the axial distance normalized by the effective diameter d_*, which is the boundary-layer thickness corrected diamter. For $d_*/d > 0.9$ this scaling results in a good collapse of the profiles.

Fig. 4.11 also shows the importance of appropriate modeling when comparing to experimental data, in particular for model validation. It is a simple task to adjust boundary conditions such that numerical simulations and experimental data coincide given such a wide variation in the velocity profile—if 'validation' is conducted inappropriately without modeling the important details of the experimental setup.

4.1.3 Summary

Good qualitative and quantitative agreement is observed between laboratory measurements and the simulation of corresponding laboratory phenomena. Both simulation and measurements provide accurate velocity profiles along the axis. In addition to comparisons along the axis, a comparison of streamlines and particle streaks provides other criteria. Based on these simulation data and experimental data, a new error function model on the axial velocity was obtained that describes axial velocity profile at range of Reynolds numbers and plate-nozzle separations uniformly. In contrast, the one-dimensional model disagrees with the experimental data and the two-dimensional simulation data if a plug-flow boundary condition is employed. However, since the error function describes the axial flow field well enough, it is possible to use it to supply an approximate boundary condition to the one-dimensional model as follows:

1. Carry out a series of 2D simulations to determine the nozzle characteristics used in the experiment and obtain Re-d^* relations.

2. Use the error-function fit to determine the centerline velocity profile.

3. Use the fit to obtain the boundary conditions for 1D model.

4. Conduct the 1D simulation.

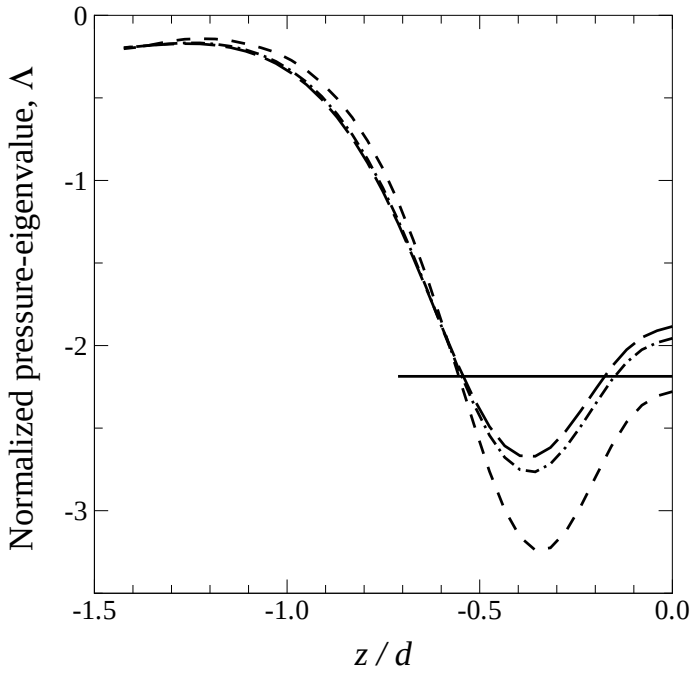


Figure 4.6: Comparison of the radial-pressure eigenvalue profile at several radial locations, $r/R = 0$ (long-dashed line), $z/R = 0.2$ (dash-dotted line), and $r/R = 0.5$ (dashed line), to that of the one-dimensional model (solid line), which is constant in both z and r.

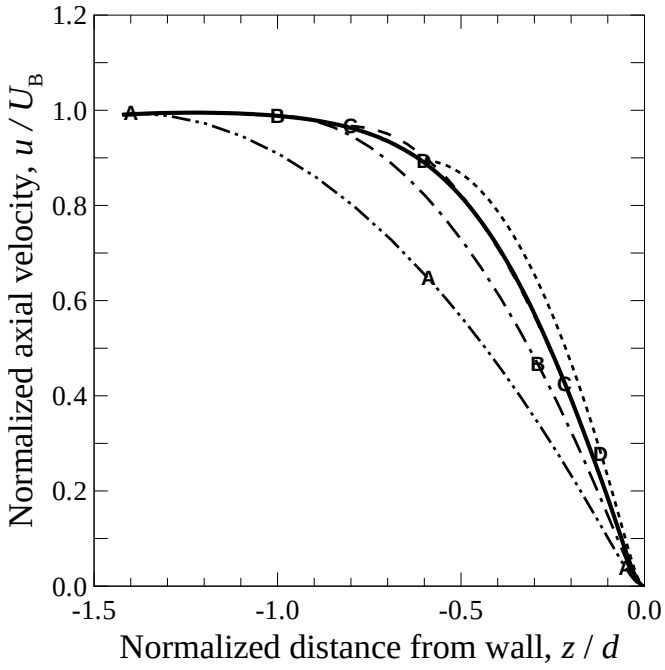


Figure 4.7: Comparison of one-dimensional numerical results with plug-flow boundary conditions with varying l/d: 1.4 (A), 1.0 (B), 0.8 (C), and 0.6 (D). The two-dimensional numerical result is shown as a reference. Plug-flow boundary conditions cannot predict the axial velocity profile at any separation distance. Although the two-dimensional result shows the velocity gradient is nearly zero at $z/d = 1.4$, the one-dimensional model cannot follow the trajectory of the two-dimensional simulation result. This is a manifestation of the uniform pressure eigenvalue assumption.

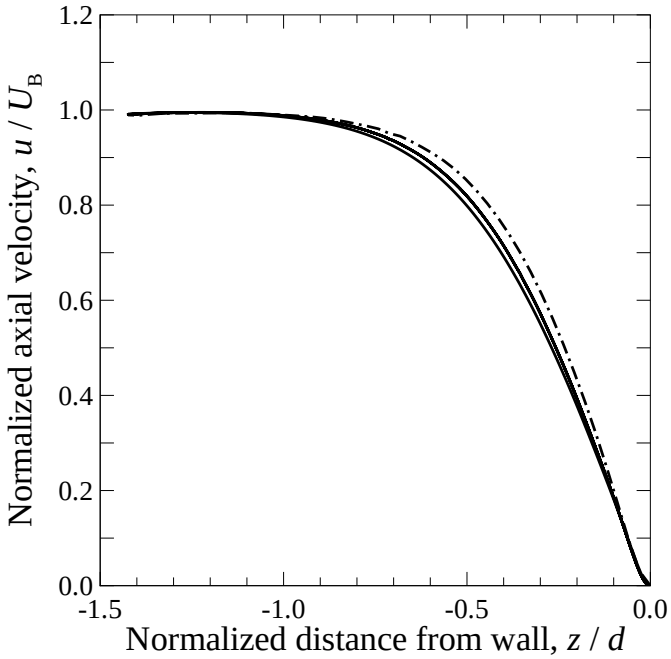


Figure 4.8: Axial velocity profiles from two-dimensional simulations versus axial distance from plate normalized by the nozzle diameter d at $L/d = 1.4$ and Re =400 (dash-dotted line), 700 (dotted line), and 1400 (solid line)

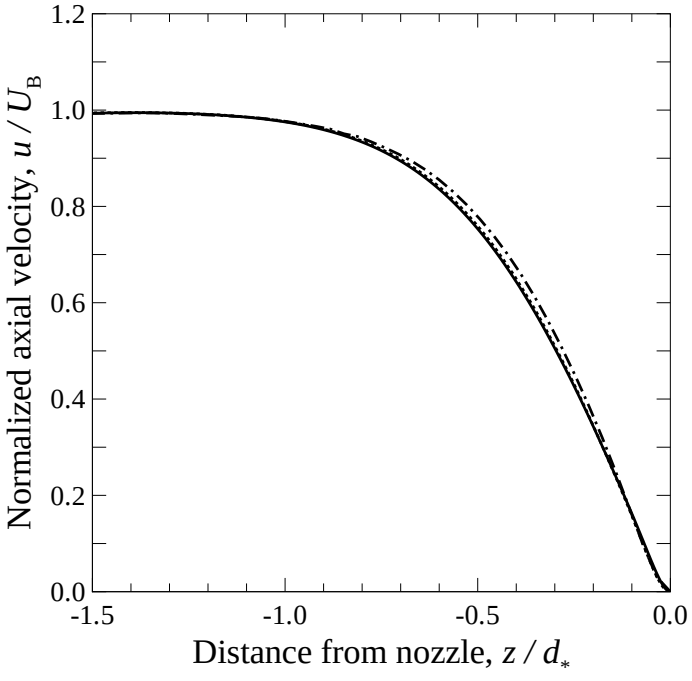


Figure 4.9: Two-dimensional simulation velocity profiles versus axial distance from the plate, normalized by the effective diameter d_* at L/d=1.4 and Re=200 (long-dashed line), 400 (dash-dotted line), 700 (dotted line), and 1400 (solid line)

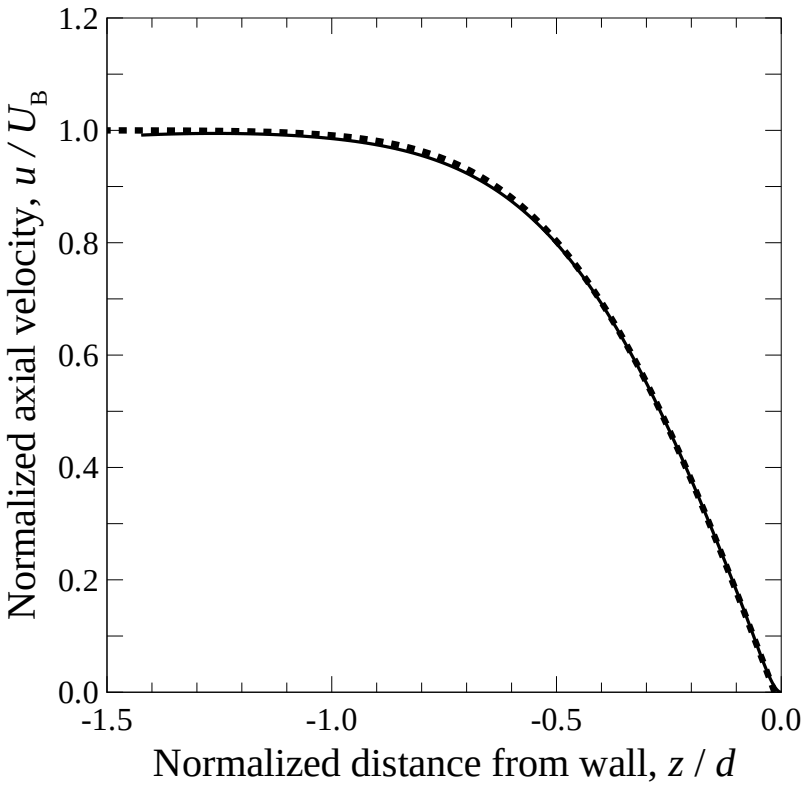


Figure 4.10: Comparison of error-function fit (dotted line) to two-dimensional simulation data (solid line) at $Re = 1400$

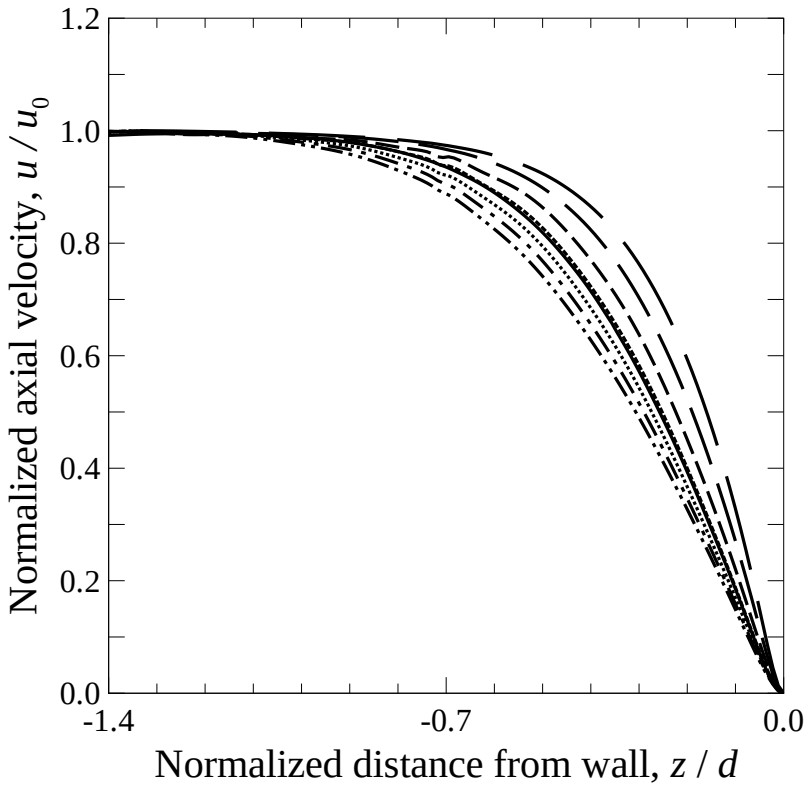


Figure 4.11: Simulated velocity profiles at $Re = 700$ and $L/d = 1.4$ for variable nozzle-exit velocity profiles: Parabolic ($d_*/d = 0.71$, long dash), hyperbolic-tangent profiles with $d_*/d = 0.76$ (medium dash), $d_*/d = 0.82$ (dash), $d_*/d = 0.87$ (short dash), $d_*/d = 0.91$ (dot), $d_*/d = 0.95$ (dash-dot), and top-hat ($d_*/d = 1.0$, dash-dot-dot) profiles. Simulation with nozzle at the same condition is plotted with a solid line, indicating that the nozzle used in the experiments by Bergthorson *et al.* (2005a) and Bergthorson (2005) produces $0.87 < d_*/d < 0.91$.

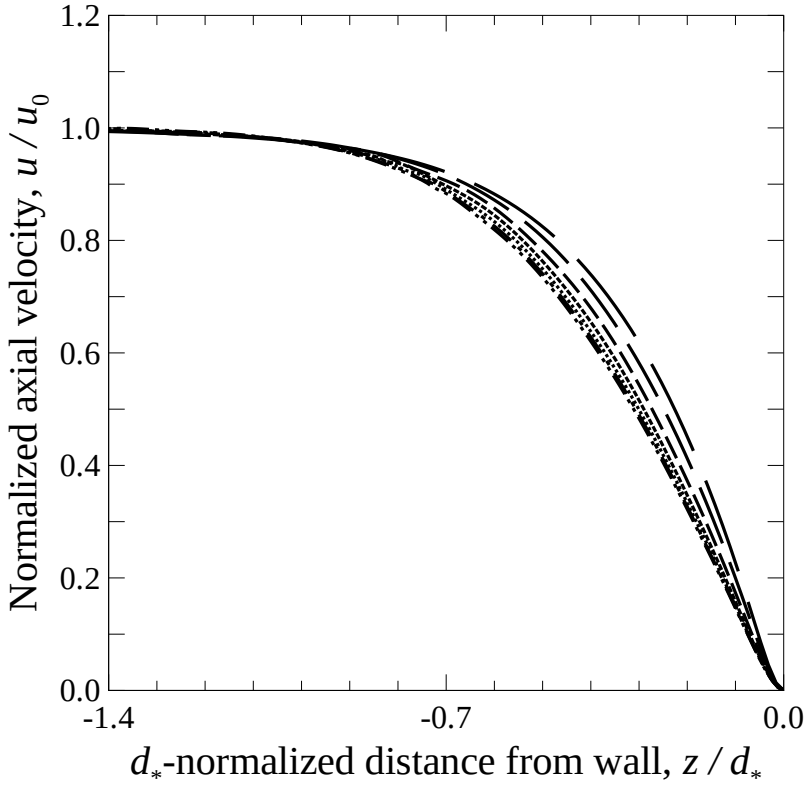


Figure 4.12: Simulated velocity profiles versus axial distance from the plate, normalized by the effective diameter d_* at $Re = 700$ and $L/d = 1.4$, for variable nozzle-exit velocity profiles. The last four curves are now nearly indistinguishable, indicating the benefit of a high contraction area-ratio nozzle. (Legend are as in Fig. 4.11.)

4.2 Steady stagnation flames

4.2.1 Introduction

When the flow supports chemical reactions, the details of each reaction must be modeled. There is more than one model for hydrocarbon combustion available. Perhaps the most widely accepted model of methane combustion is GRI-Mech 3.0. Although methane is the simplest hydrocarbon and its combustion is relatively well understood among hydrocarbon fuels, there are still uncertainties and modeling its chemical kinetics is far from satisfactory, mostly because of the large number of reactions involved. For example, GRI-Mech 3.0, less nitrogen reactions (model HC1 in Table 1.1), features 217 reactions, whereas its hydrogen reaction subset contains only 28 reactions.

Although widely used, these reaction mechanisms have not been completely validated. A realistic simulation of a laboratory flame has been computationally expensive and validation attempts have been difficult. Therefore simulations of hydrocarbon flames reported to date can be categorized into two types: ones that employ full chemical kinetics models such as GRI-Mech 3.0 along with a one-dimensional fluid mechanical model proposed by (Kee *et al.*, 1988), and ones that use a multi-dimensional Navier-Stokes model of fluid mechanics with a reduced chemistry model (Smooke *et al.*, 1986, 1992). For example, Egolfopoulos (1994) used one-dimensional model and the EDL92 mechanism (see Table 1.1) to study radiation effects in an opposed-jet premixed and non-premixed flames. To the author's knowledge, no multi-dimensional simulations of hydrocarbon stagnation or opposed-jet flames were reported, however, there are several multi-dimensional simulations of laminar hydrocarbon flames. Bennett *et al.* (1999) simulated a steady axisymmetric Bunsen flame using GRI-Mech 1.2 (HC2 in Table 1.1). Regardless of the type of problems, many kinetic mechanisms have been studied mostly within the context of the one-dimensional hydrodynamic model, with numerical results subsequently compared to experiments. Of course, the fluid mechanical part of the one-dimensional model is also approximate, as was shown in the case of cold flow, and comparisons within such a oversimplified framework can be misleading.

As explained earlier, an efficient algorithm has been developed to simulate hydrocarbon flames in this study. To further reduce computational time required before the simulated flame reaches to the steady state, the simulations are conducted in three phases. Phase I relies on the one-dimensional simulation using Cantera. This provides an approximate flame

location, as well as desired mesh distribution so that the Phase II computational domain can have a cluster of elements near that region. In addition, it provides a good initial condition to the Phase II simulation, as described in Appendix B. The Phase II simulation is the axisymmetric two-dimensional simulation with a truncated domain. The computational domain contains a stagnation region and some of the post-flame and entrainment regions only. No nozzle-wall proximity effect is considered in this case. All Phasse II computations reported in the present study are recorded in Table 4.2. Finally, Phase III is the "real" simulation that contains the interior of the nozzle and the enclosure of the nozzle-wall assembly, within which nozzle-wall proximity effects as well as entrainment is appropriately captured in the simulation, and models the experiments documented by Bergthorson *et al.* (2005b), Bergthorson & Dimotakis (2007), and Bergthorson (2005). All Phase III simulations are recorded in Table 4.3.

All reacting flow simulations reported in this study have a unique case number. The first character signifies the type of the domain: A for 1-D model, B for 2-D Phase II model, and C for 2-D Phase III model, followed by three-digit numbers that corresponds to the equivalence ratio—070 should read $\Phi = 0.70$. The number after the dash is a unique identification number among each setup. All simulations reported in this study use the GRI-Mech 3.0 model, less nitrogen reactions (model HC1 in Table 1.1). It was confirmed independently that the nitrogen chemistry do not alter the numerical results by comparing one-dimensional results using the HC1 and HCN1 mechanisms in Table 1.1.

In their study of laminar flame speed, Wu & Law (1984) suggested the use of stretched flames to determine the laminar flame speed in the limit of flame stretch going to zero and demonstrated the use of stagnation flames for this purpose. Since then, this extrapolation-based laminar flame speed determination has been commonly used. Later Tien & Matalon (1991) showed that the flame-stretch to flame-speed relationship can vary depending on how reference locations are selected, and they resolved inconsistency and confusion in the literature about how to extrapolate to obtain laminar flame speed. Such extrapolated laminar flame speeds are reported, for example, in Egolfopoulos *et al.* (1997), Dong *et al.* (2002), and Hirasawa *et al.* (2002), and comparisons between experimentally determined laminar flame speed (through extrapolation) and simulation of freely propagating flame (using 1-D codes such as CHEMKIN / PREMIX) have been made. Bergthorson *et al.* (2005b) were probably the first to attempt estimates of modeling errors in GRI-Mech 3.0

Table 4.2: Phase II simulation data. (See page 78 for the definitions of symbols.)

Case	Φ	σ (1/s)	Γ (1/s)	$S_{\mathrm{u,ref}}$ (m/s)	z_{Flame} (m)	κ (1/m)	Figure
B070-2	0.70	67.01	118.80	0.241709	-0.004353	184	C.1
B070-7	0.70	82.25	111.69	0.242096	-0.003941	114	C.2
B070-12	0.70	97.80	107.24	0.2437	-0.003650	77	C.3
B070-17	0.70	111.79	107.91	0.2454	-0.003360	48	C.4
B070-22	0.70	123.22	114.36	0.2471	-0.003100	44	C.5
B070-27	0.70	135.84	119.80	0.2479	-0.002836	16	C.6
B070-30	0.70	142.67	124.37	0.2476	-0.002681	13	C.7
B080-17	0.80	136.98	138.32	0.3212	-0.004047		C.8
B080-46	0.80	223.46	178.60	0.3346	-0.002769		C.9
B090-35	0.90	184.13	173.83	0.3820	-0.003966	30	C.10
B090-50	0.90	296.77	228.21	0.3942	-0.002695	13	C.11
B090-106	0.90	342.38	258.72	0.3989	-0.002322	8	C.12
B090-112	0.90	406.32	312.93	0.4015	-0.001855	4	C.13
B100-23	1.00	228.38	200.92	0.4201	-0.003740		C.14
B120-10	1.20	171.81	189.01	0.3707	-0.003760	43	C.15

Table 4.3: Phase III simulation data. (See page 78 for the definitions of symbols.)

Case	Φ	σ (1/s)	Γ (1/s)	$S_{\mathrm{u,ref}}$ (m/s)	z_{Flame} (m)	κ (1/m)	Figure
C070-4	0.70	104.01	67.62	0.2365	-0.004421	-49	C.16
C070-11	0.70	112.20	71.20	0.2386	-0.004133	-55	C.17
C070-8	0.70	115.91	73.46	0.2395	-0.003997	-56	C.18
C070-9	0.70	127.60	81.19	0.2423	-0.003633	-59	C.19
C090-5	0.90	209.33	89.76	0.3734	-0.004936	-44	C.20
C090-2	0.90	238.55	116.42	0.3819	-0.004224	-37	C.21
C090-4	0.90	246.01	122.25	0.3848	-0.004103	-31	C.22
C090-3	0.90	253.51	128.79	0.3862	-0.003952	-30	C.23
C120-1	1.20	229.77	126.68	0.3512	-0.003694	-29	C.24

without extrapolating to a stretch-free state. In their approach, instead of flame speed, flame location was used as a surrogate for flame strength, and they studied flames at different equivalence ratios using appropriate dilutions to allow the flame to be realized at about the same strain rate.

Although in their original paper, Wu & Law (1984) used the word 'stretch,' what they used as a measure of flame stretch was a strain-rate (normal strain) upstream of the flame where the effect of heat release is minimal. Flame stretch, as defined later, will be accepted as representing a lateral-strain rather than normal-strain. Because in the case of idealistic

potential flow solution of stagnation flow, strain and stretch are related and differ only by a factor of two (Law & Sung, 2000), if measured at the same location and dilatation is excluded. This seems to be the cause of mixed use about 'stretch' and 'strain' in the literature when, in many cases, both terms mean the same quantity—a flame-normal strain-rate, $\mathbf{n} \cdot \nabla \mathbf{u} \cdot \mathbf{n}$, where $\mathbf{n}$ is a normal vector of a flame front. Of course, when defined at the same location, strain and stretch are related through the continuity equation. For example, along the axis, $\sigma_n + 2\sigma_t = -\partial\rho/\partial z$ where $\sigma_n = \partial u/\partial z$ and $\sigma_t = \partial v/\partial r$

4.2.2 Results and discussion

4.2.2.1 Validation of the one-dimensional model

The uniform-pressure eigenvalue assumption and the inappropriateness of plug-flow boundary conditions were shown to lead to erroneous results for the cold flow case. The behavior of the same one-dimensional model when chemical reactions are present is investigated here. The computational domain used for this study is shown in Fig. 4.13. The one-dimensional numerical solution for the same separation distance at the same flow condition is obtained *a priori* for each two-dimensional simulation, and the solution is extrapolated in the radial direction to obtain an initial condition to the two-dimensional model (see Appendix B). The inflow velocity condition, as well as the entrainment flux, is specified at the nozzle exit. The wall is isothermal and kept to $T_{\text{wall}} = 300\,\text{K}$ throughout the study. The outflow boundary condition is applied at the outermost boundary at $r = 2.0\,\text{cm}$. In the following analysis, velocity and other quantifies are plotted only along the axis unless otherwise noted.

Shown in Fig. 4.14 is an axial velocity profile comparison along the axis between the two-dimensional model and the one-dimensional model (Cantera), at the equivalence ratio of $\Phi = 0.70$. The velocity profile deviates as it approaches the flame when the plug-flow boundary condition is used, and there is a difference in the prediction of flame location between two-dimensional and one-dimensional models. This difference is attributable to the uniform pressure-eigenvalue assumption in the one-dimensional formulation. However, this can be overcome by specifying the boundary condition in the middle of the nozzle-flame proximity with the velocity boundary condition values taken directly from the two-dimensional simulation, as discussed by Bergthorson *et al.* (2005b) for comparisons between experiments and one-dimensional simulations, and as shown in Fig 4.15. (Hereafter, this

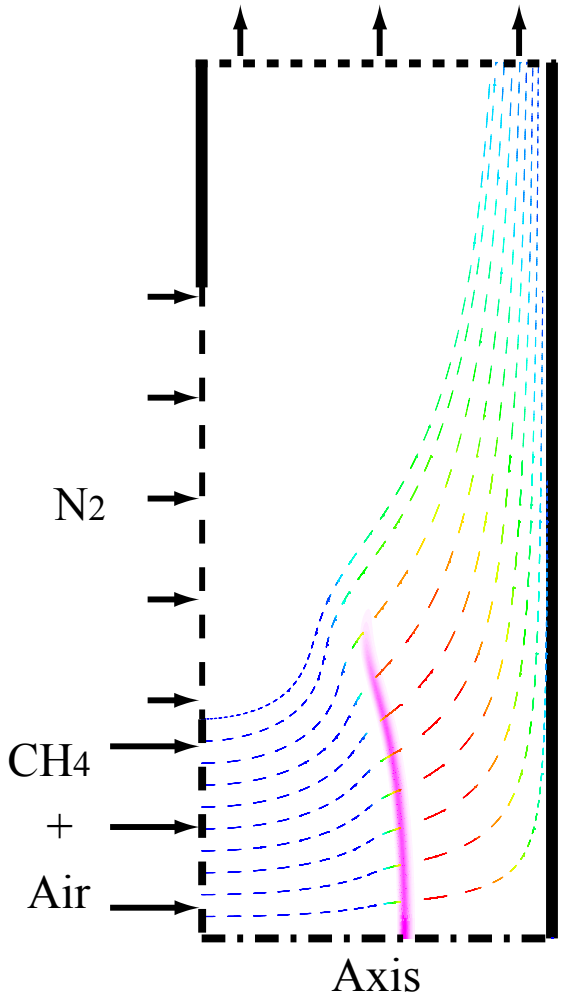


Figure 4.13: A computational domain used for the Phase II numerical studies and a typical realization of the flow field (streamlines shown as dashed lines), temperature field (color of the streamlines), and the flame (mass fraction of CH shown in pink). The actual allocation of elements within the domain varies for each case.

short domain case with a finite spreading rate boundary condition is denoted as 1D-s.) It indicates the plug-flow boundary condition cannot be used for any practical comparison with experimental data or multidimensional simulations, even when there is no velocity gradient at the nozzle exit, which is consistent with previous findings in the case of non-reacting flow (Bergthorson *et al.*, 2005a). To illustrate the discrepancy between two- and one-dimensional models, Fig. 4.16 presents the pressure-eigenvalue profiles. In one-dimensional models, this is assumed to be a constant, as indicated in the figure (dashed line). The pressure eigenvalue profile has a large spike near the flame front. This is in contrast to earlier hydrogen flame simulations reported by Frouzakis *et al.* (1998) and provides another reason that renders hydrocarbon flame simulation challenging. This spike is mainly because of the thinness of the flame as well as the curvature of the flame front. The magnitude of the pressure eigenvalue affects the prediction of the spreading rate, $V(= \partial v/\partial r)$, in the one-dimensional model and results in an overprediction or underprediction of the spreading rate in the near- and post-flame region, as shown in Fig. 4.17. On the other hand, the location and the size of the spike has no discernible effect on the prediction of flame location in the one-dimensional model, as shown in Fig. 4.15. This implies that the pressure eigenvalue profile between nozzle and the flame is an important factor in determining flame location, and must somehow be also accounted for in one-dimensional formulations, if high accuracy is of importance.

As the volume flux through the inlet boundary is increased, the flame moves close to the stagnation wall. The pressure-eigenvalue profile changes as the flame moves, as shown in Fig. 4.19. The depth of the spike reduces gradually as the flame moves to the wall. This is primarily because the flame loses its curvature as the strain-rate increases.

To discuss the flame behavior, let us define several important parameters about flames used in the present study: strain (σ), stretch (Γ), reference flame speed ($S_{\mathrm{u,ref}}$), location (z_{flame}), and curvature (κ). In this study, the flame is defined as a set of points that maximizes the mass fraction of CH along each streamline. Presently, analysis is conducted only along the axis, and therefore, z_{flame} is a point such that $Y_{\mathrm{CH}}(z_{\mathrm{flame}}, 0)$ is maximum along the axis. Strain-rate, σ, is defined by the maximum slope upstream of the flame, z_{flame}, which is an inflection point of axial velocity curve, whereas the reference flame speed is the local minimum velocity upstream of the flame. The flame stretch is a lateral strain at the flame front, $\partial v/\partial r(z_{\mathrm{flame}}, 0)$, and is defined more precisely later. These definitions

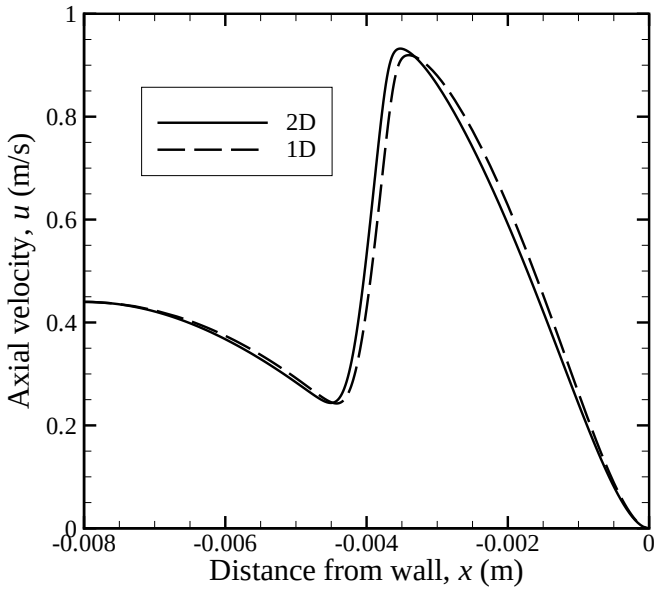


Figure 4.14: Comparison of two-dimensional and one-dimensional solutions with the plug-flow boundary conditions to one-dimensional model. Both models employ the same domain length (0.008 m) in the axial direction. Even with the same zero-slope boundary condition specified to both two- and one-dimensional models, the one-dimensional model fails to follow the axial velocity profile of the two-dimensional model.

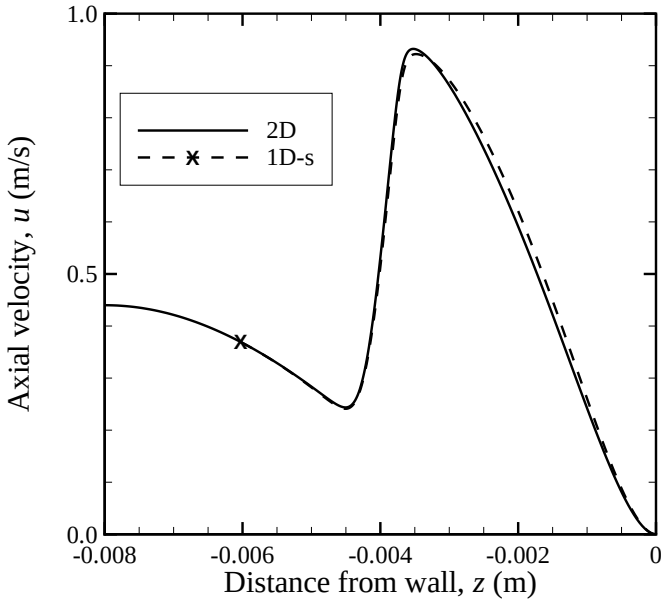


Figure 4.15: Comparison of two-dimensional and one-dimensional solutions with the boundary conditions to one-dimensional model specified at $z = -0.006$. Better agreement between the one-dimensional model and the two-dimensional model can be seen in this case, compared to the plug-flow boundary condition (Fig. 4.14).

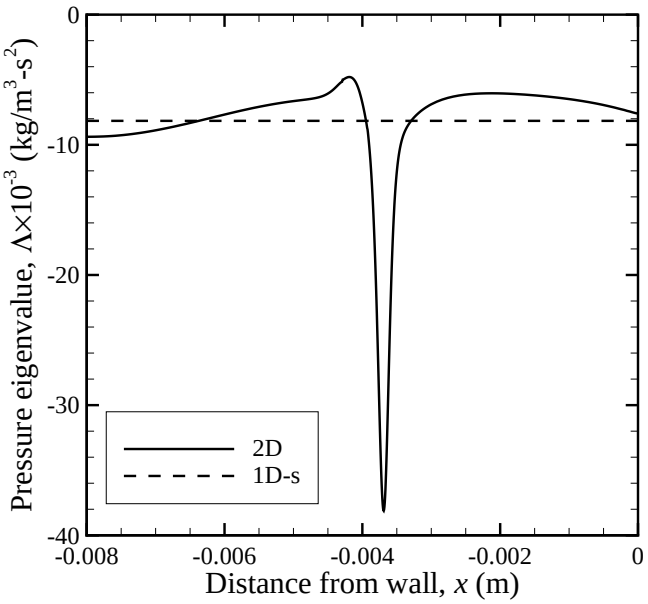


Figure 4.16: Pressure eigenvalue profile along the axis in a methane-air flame ($\Phi = 0.7$). Note the sharp spike at the flame front when the correct pressure field is obtained, which contrasts with a uniform profile assumed in the one-dimensional model. (Legend as shown within the plot.)

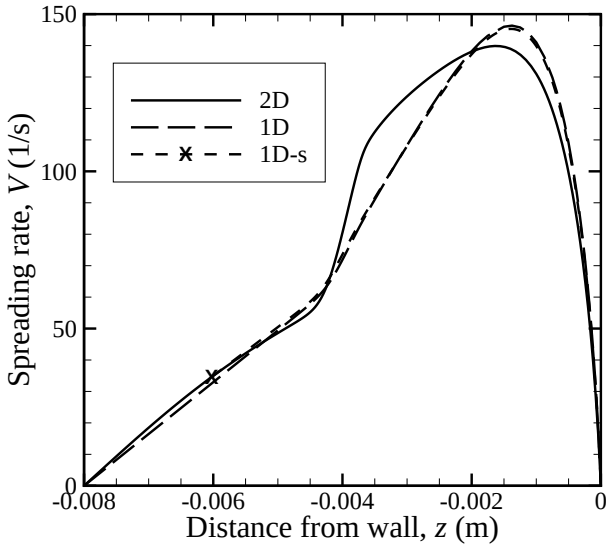


Figure 4.17: The spreading rate, $\partial v/\partial r$, profile along the axis in a methane-air flame ($\Phi = 0.7$). Within the reaction zone, the spreading rate changes, which is an effect that is not incorporated in the one-dimensional model as a consequence of the uniform pressure-eigenvalue assumption.

are schematically shown in Fig. 4.18. It should be pointed out, in this convention, the locations at which strain is defined and stretch is defined are different. The former is defined somewhere in the early part of preheat zone while the latter is defined at the flame front—within a thin reaction zone.

The local flame curvature on the symmetric axis can be written as,

$$\kappa = 2/\mathcal{R}_{\text{Flame}}, \tag{4.5}$$

where $\mathcal{R}_{\text{Flame}}$ is the radius of curvature of the flame front and is given by (Gray, 1997)

$$\frac{1}{\mathcal{R}_{\text{Flame}}} = \frac{g_{zz}g_r^2 - 2g_{zr}g_z g_r + g_{rr}g_z^2}{(g_z^2 + g_r^2)^{3/2}}, \tag{4.6}$$

where $g(z,r) = 0$ defines the flame front and subscripts denote its derivatives in the above expression. Since the CH radical is used as a marker in the present study, $g(z,r) \equiv \partial Y_{\text{CH}}/\partial z$ can be used. Furthermore, $g_r = 0$ along the axis. This simplifies the expression for curvature to:

$$\kappa = 2\frac{g_{rr}}{g_z}. \tag{4.7}$$

The factor of two is because of the azimuthal symmetry (*e.g.*, Candel & Poinsot, 1990) for axisymmetric flames.

Curvature is positive when the flame is concave toward the unburnt gas and negative when the flame is convex toward the unburnt gas. Phase II simulations always produced positively curved flames. The shapes and the curvature of flames at various conditions can be observed in the flame-simulation data in Appendix C. When curvature is small, the one-dimensionl simulation with the slope-matched boundary condition (1D-s) works well, as shown in Figs. 4.20 and 4.21. However, when flame curvature is large, even the 1D-s case does not follow the axial velocity profile obtained by two-dimensional simulation, as shown in Figs. 4.22 and 4.23. It is imperative to keep flame curvature as small as possible if a comparison to the one-dimensional model is to be meaningful.

Shown in Figs. 4.24 and 4.25 are the axial velocity and the spreading rate at the equivalence ratio of $\Phi = 0.90$. The profile for the $\Phi = 1.20$ case is plotted in Figs. 4.26 and 4.27. In both of these cases, the one-dimensional model predicts a stronger flame than is predicted by the two-dimensional model. The implications will be discussed later.

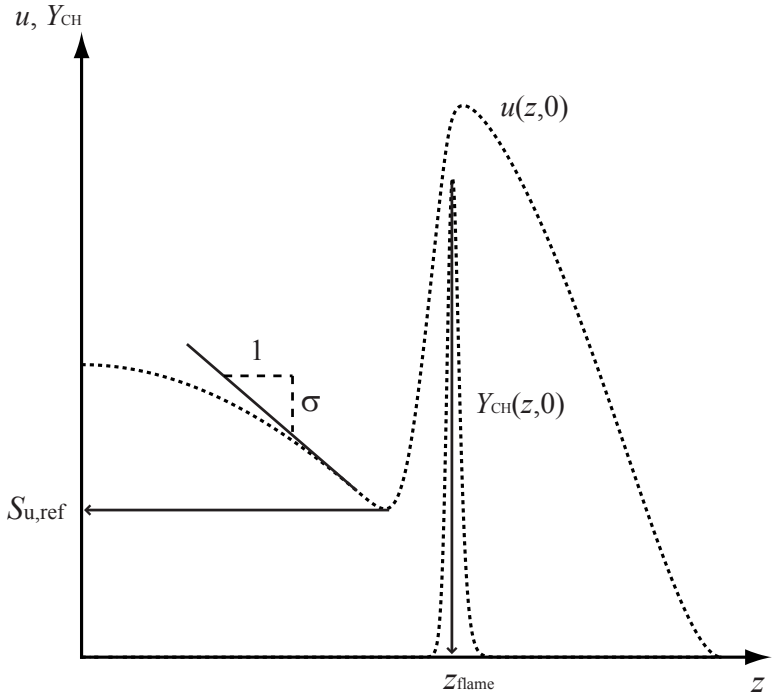


Figure 4.18: Definitions of some of important parameters describing flames: σ: strain-rate, $S_{\mathrm{u,ref}}$: Reference flame velocity, and z_{flame}: flame location are shown here. κ, flame curvature, and Γ, flame stretch, are defined at $z = z_{\mathrm{flame}}$ and $r = 0$.

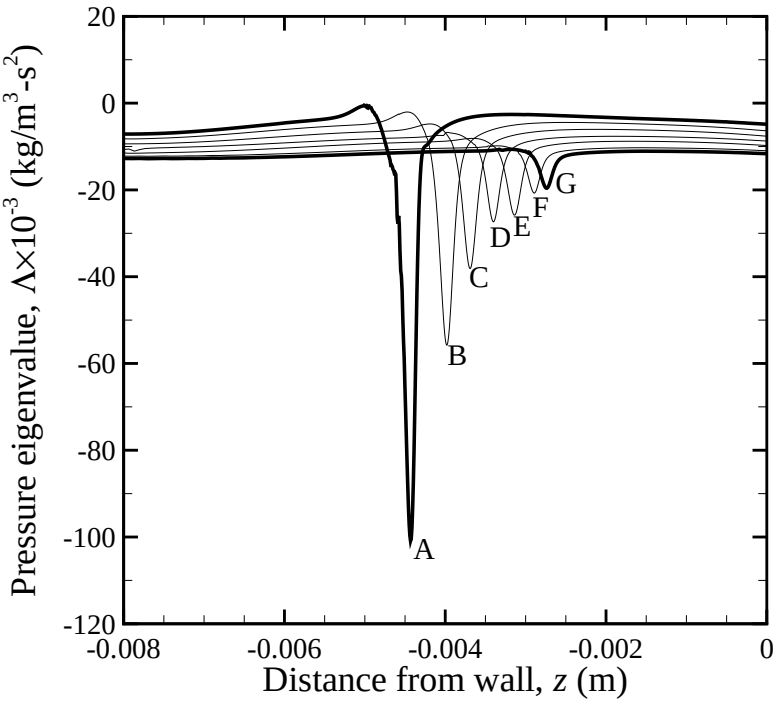


Figure 4.19: Variation of pressure-eigenvalue profile as flame strain-rate changes. Note the depth of the spike decreases gradually as the flame loses its curvature. (A): Case B070-2; (B): Case B070-7; (C): Case B070-12; (D): Case B070-17; (E): Case B070-22; (F): Case B070-27; (G): Case B070-30. See Table 4.2 for parameters for each case.

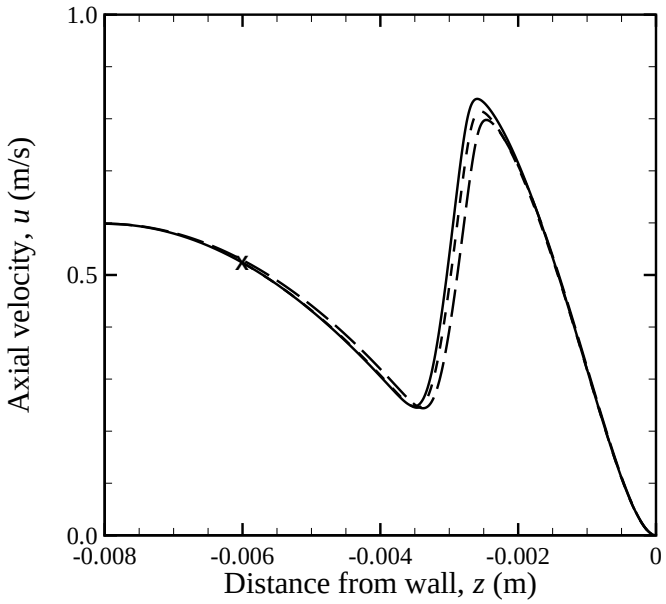


Figure 4.20: Comparison of the axial velocity profile between the one-dimensional model (Case A070-30) and the two-dimensional model (Case B070-30) using the same kinetics mechanism at $\Phi = 0.70$. In this case, the pressure-eigenvalue profile has the smallest peak in absolute value, and the difference between the one-dimensional model and the two-dimensional model is small. 2-D (solid line), 1-D with plug-flow BC (long dashed line), and 1-D with boundary conditions taken from 2-D simulation (short dashed line with X denoting the location where the boundary condition is specified)

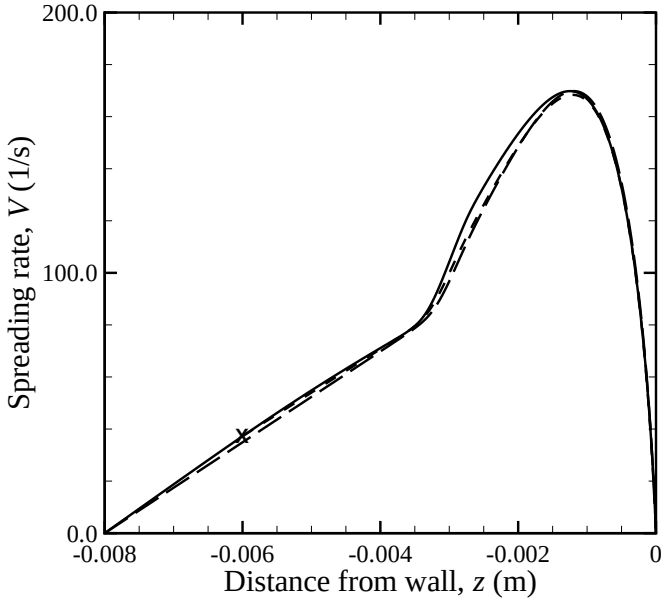


Figure 4.21: Comparison of the spreading rate (radial velocity gradient) profile between the one-dimensional model (Case A070-30) and the two-dimensional model (Case B070-30) using the same kinetics mechanism for $\Phi = 0.70$. Again, the difference is smaller compared to the earlier case when a large pressure eigenvalue led to a large discrepancy in the spreading rate. (Legend as in Fig. 4.20.)

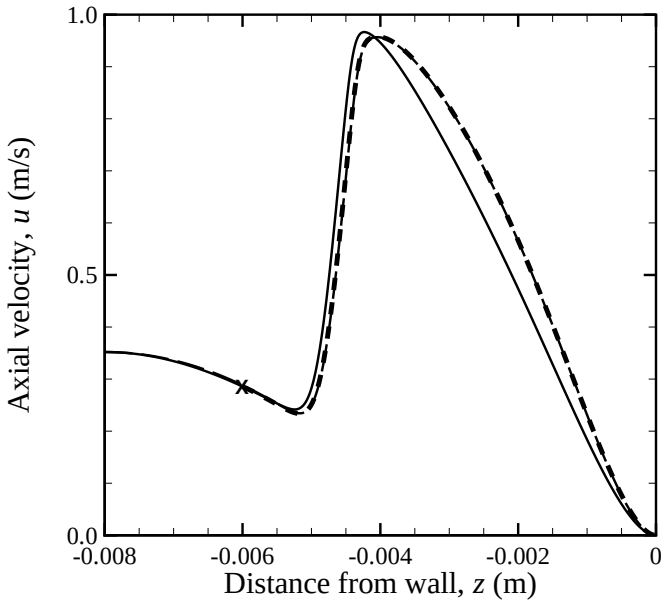


Figure 4.22: Comparison of two-dimensional (Case B070-2, solid) and one-dimensional (Case A070-2) solutions with two different boundary conditions: Plug-flow (short dash) and velocity-gradient specified with truncated domain (long dash). 'X' denotes the location where the boundary condition is specified for the truncated-domain case. This is the flame with the smallest strain-rate realized in this study. The equivalence ratio is 0.70. In this case, the flame is curved so strongly (see Fig. C.1 in Appendix C) that flame location cannot be predicted by the one-dimensional solution, even with the correct boundary conditions given by the two-dimensional solution.

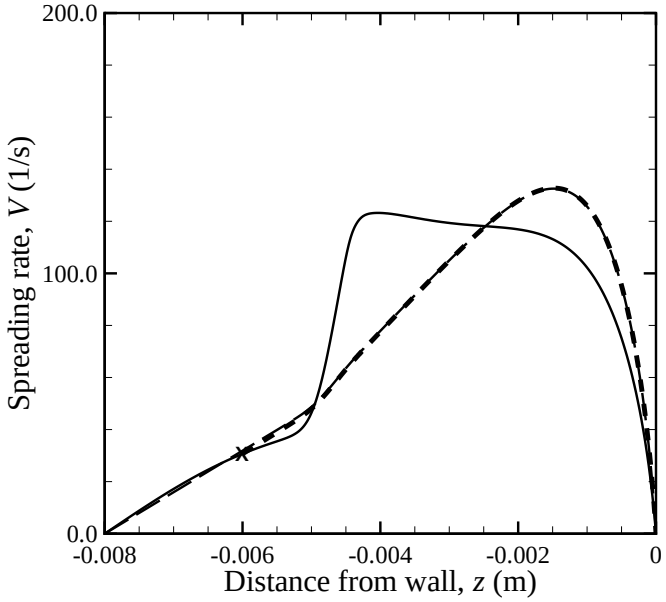


Figure 4.23: Comparison of the spreading rate, $V(= \partial v/\partial r)$, along the axis between two-dimensional (Case B070-2) and one-dimensional (Case A070-2) solutions. (Legend as in Fig. 4.22.)

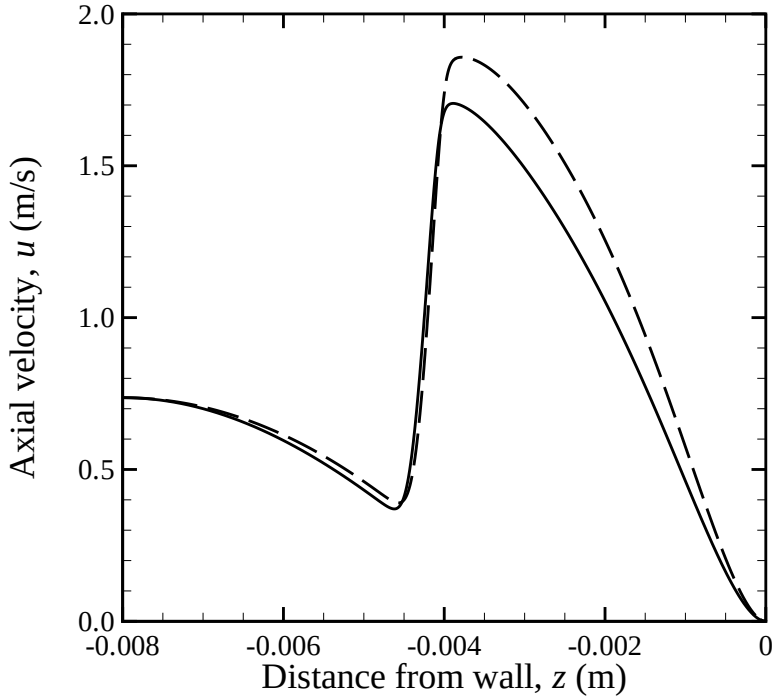


Figure 4.24: Comparison of the axial velocity profile between the two-dimensional model (Case B090-35) and the one-dimensional model (Case A090-35), using the same kinetics mechanism, for $\Phi = 0.90$. (Legend as in Fig. 4.20.)

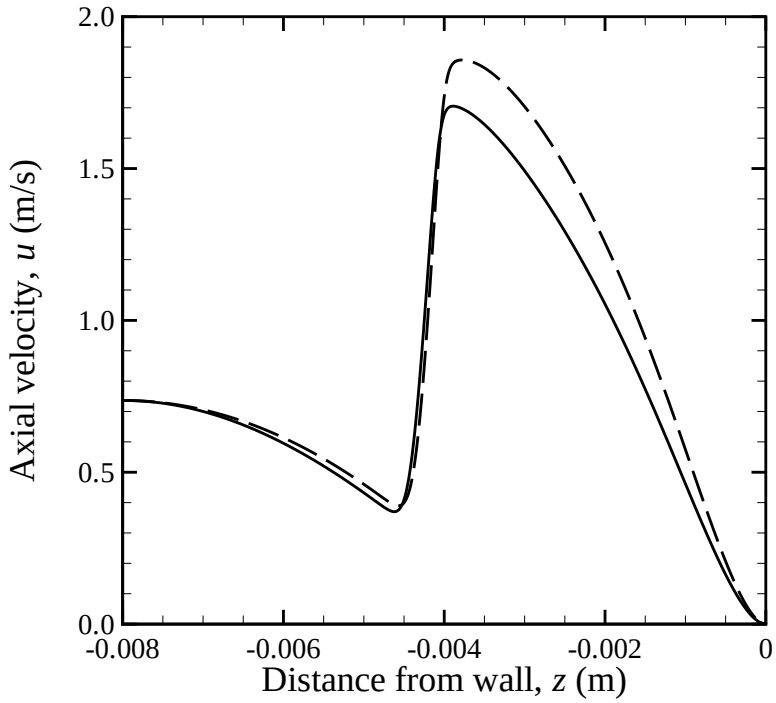


Figure 4.25: Comparison of the spreading rate (radial-velocity gradient) profile between the two-dimensional model (Case B090-35) and the one-dimensional model (Case A090-35), using the same kinetics mechanism, for $\Phi = 0.90$. (Legend as in Fig. 4.20.)

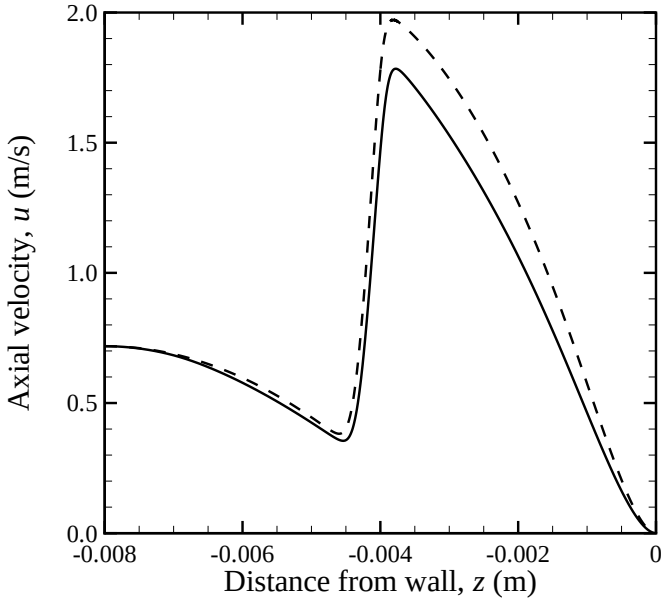


Figure 4.26: Comparison of the axial velocity profile between the two-dimensional model (Case B120-10) and the one-dimensional model (Case A120-10), using the same kinetics mechanism, for $\Phi = 1.20$. (Legend as in Fig. 4.20.)

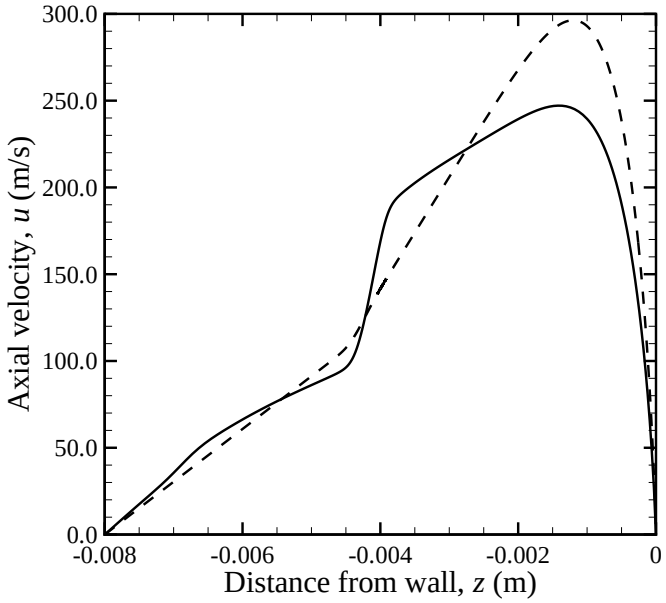


Figure 4.27: Comparison of the spreading rate (radial velocity gradient) profile between two-dimensional model (Case B120-10) and one-dimensional model (Case A120-10), using the same kinetics mechanism, for $\Phi = 1.20$.

4.2.2.2 Simulation of the stagnation flame in the laboratory and validation of the chemistry model

The discussions in this section focus on the capability and limitation of methane combustion models, such as GRI-Mech 3.0. As explained above, it is not a computational code that must be validated, but a model in the code that must be validated against experimental data to measure the modeling error. This is, to our knowledge, the first attempt to investigate the validity of a detailed methane chemical-kinetic model using multi-dimensional simulations with a realistic transport model that takes into account differential-diffusion effects.

The computational domain used for this study is shown in Fig. 4.28. It includes the interior and the exterior of the nozzle-plate assembly at the dimensions specified in the corresponding laboratory setup (Bergthorson, 2005; Benezech *et al.*, 2006). The combustible mixture is introduced at the bottom of the nozzle (the lower-left end) and a moderate amount of air is introduced at the bottom of the exterior between nozzle and the enclosure (the upper-left end) to accomodate the entrainment requirements of the jet flow. To stabilize the flame, in both experiments and simulations, a small amount of nitrogen is introduced from an outer nozzle, for which an exit velocity profile is specified in the simulation since the nozzle-wall proximity effect is not important in this region. All fluids entering the computational domain exit from the side of the stagnation plate, which can be seen at the upper-right end in the figure.

Figs. 4.29 through 4.33 show the contour plots of the mass fraction of CH radical, the pressure field, the axial velocity, the divergence constraint, and the heat-release rate for $\Phi = 0.70$ (Case 11), respectively. Compared to the earlier case (Phase II), in which the velocity boundary condition is specified at the nozzle exit, this one has a dip in the flame shape, which is a consequence of the nozzle-flame proximity effect that modifies the velocity profile at the nozzle exit. This flame shape has been observed in the laboratory and provides additional qualitative comparison between experiments and simulations.

This can be seen clearly in the plot of the pressure-eigenvalue (Fig. 4.34) and that of the spreading rate (Fig. 4.35). The pressure eigenvalue profile now has a peak that is positive at the flame front rather than a negative-valued spike. This positive pressure eigenvalue creates a plateau in the spreading rate profile at around the flame front when the negative spread from the negative curvature and the positive spread from the heat release balance.

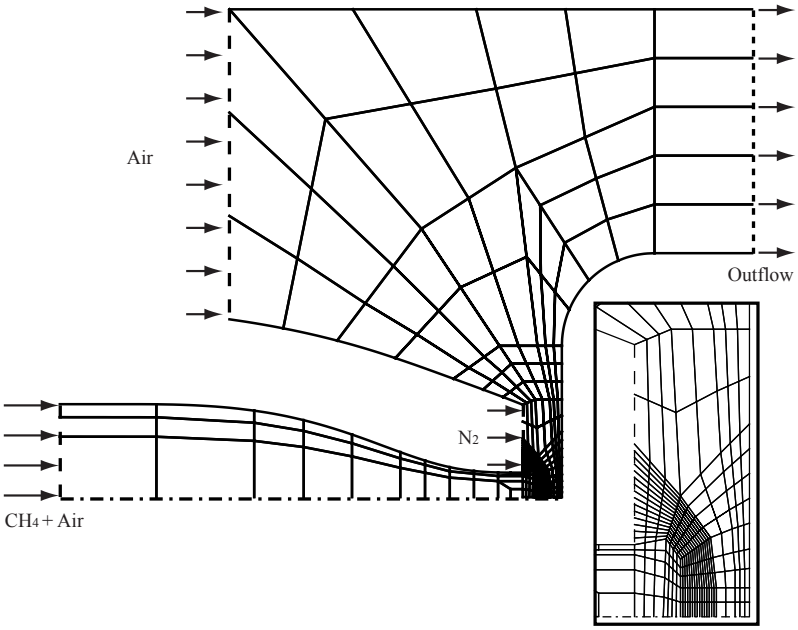


Figure 4.28: The computational domain and typical elements used for the simulation of stagnation flames of the laboratory experiments by Bergthorson & Dimotakis (2007). This computational domain contains 375 elements, and there are 12 by 12 collocation points within each element. Inflow/outflow boundaries can be recognized by the direction of the arrows. Note that the length of the arrows does not indicate the velocity magnitude. Solid lines on the boundary indicate solid isothermal walls, all of which are at $T_{\text{wall}} = 300$ K.

When the flame becomes stronger ($\Phi = 0.90$), the peak of the eigenvalue profile becomes large (Fig. 4.36) and the plateau of the spreading rate profile becomes a dip in the profile, as seen in Fig. 4.37.

As previously mentioned, the determination of laminar flame speeds is a major topic in laminar flame research. One of the main goals of this work is to conduct a numerical simulation that appropriately models the reacting flow observed in the laboratory so that a comparison of simulation and experiments can be done directly and appropriately. Shown in Fig. 4.38 is a comparison of the axial velocity profile obtained from the laboratory experiments by Bergthorson (2005) to the present simulation at an equivalence ratio of 0.70. The flow rate through the nozzle was not reported in the experiment, and was adjusted in the numerical simulation such that the nozzle exit velocity matches the first point in the measurements. The correction to the numerical data is applied to simulate the way in which the axial velocity field is measured in the laboratory, which is suggested in Bergthorson *et al.* (2005b) and Bergthorson (2005), and plotted as well to simulate PSV-measured

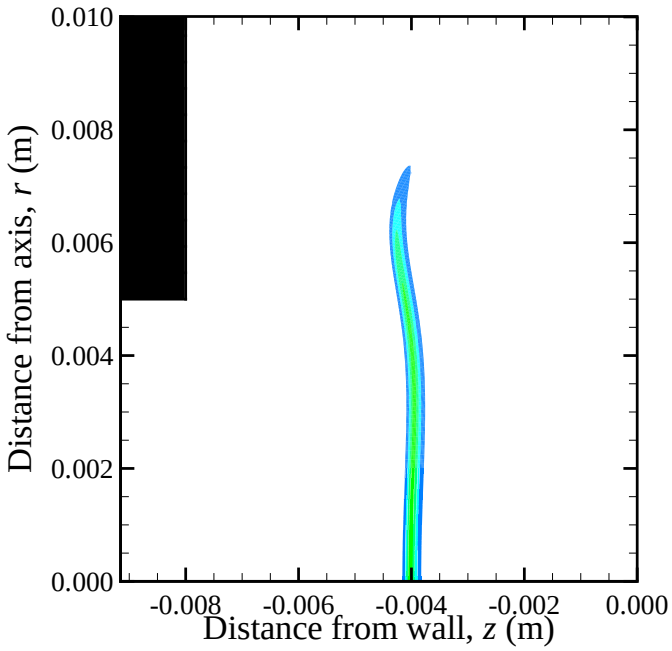


Figure 4.29: CH contour plot of lean flame ($\Phi = 0.70$). CH marks the chemiluminescence of lean flame and is used as a marker of the flame front in the laboratory. Note the flame has a dip in the middle due to nozzle-flame proximity effect. This can be seen in the laboratory.

velocity data from numerical simulation data. A *MATLAB*®script that simulates PSV is included in Appendix D of the present work. When the equivalence ratio is close to unity, a good agreement to experimental data is observed, as shown in Fig. 4.39. Qualitative agreement in flame shape can also be seen in Fig. 4.40, in which the experimental image of the flame shape through CH-PLIF imaging and computed CH contours are superimposed. Flame location and flame shape are both well-captured in the numerical result. For rich mixtures, GRI-Mech 3.0 with a two-dimensional model underpredicts inferred flame speed from that measured in a laboratory. Shown in Fig. 4.41 is a comparison of two-dimensional simulation to one-dimensional simulation. Due to lack of sufficient data in both experiments and simulations, there is no direct comparison for this case, however, from previous work (Bergthorson *et al.*, 2005b; Bergthorson, 2005), it is known that one-dimensional simulation tracks experimental data well. Provided that one-dimensional simulation with GRI-Mech 3.0 follows experimental mesurements, Fig. 4.41 implies GRI-Mech 3.0 underpredicts flame speed at this equivalence ratio of $\Phi = 1.20$.

Reference flame speeds are computed both from simulations and from measurements,

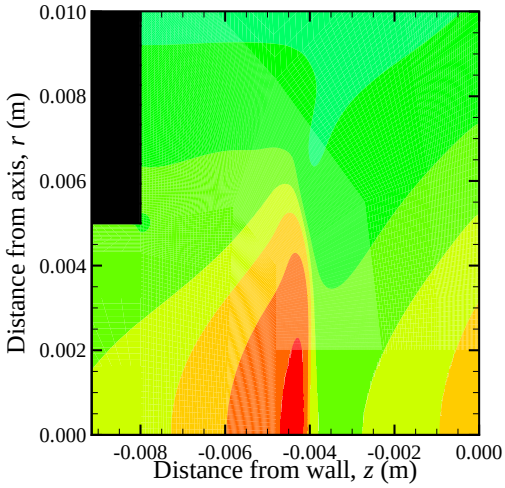


Figure 4.30: Pressure contour. Note that pressure has a large peak just upstream of the flame. The increased static pressure is fed into the nozzle, which causes the axial velocity deficit, and creates a negatively curved flame.

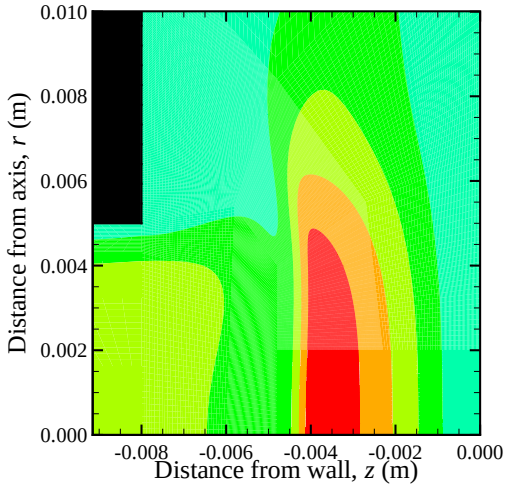


Figure 4.31: Axial velocity contours. Note curvature of iso-velocity lines in the vicinity of the flame.

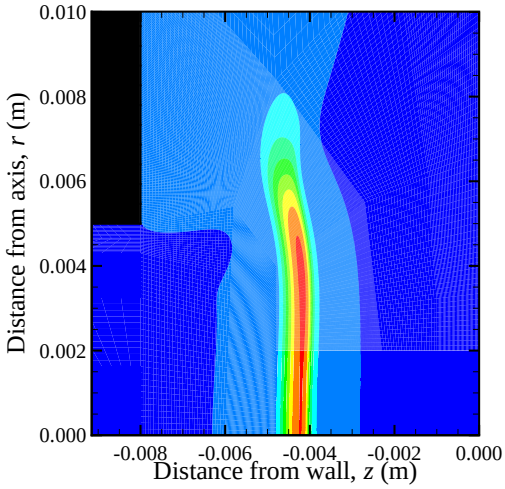


Figure 4.32: Contours of the divergence field

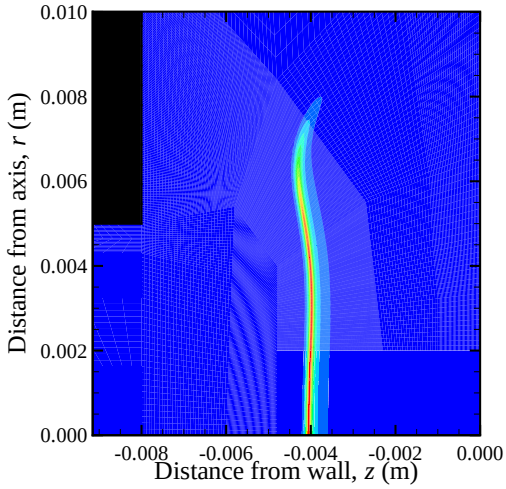


Figure 4.33: Contours of the heat-release rate

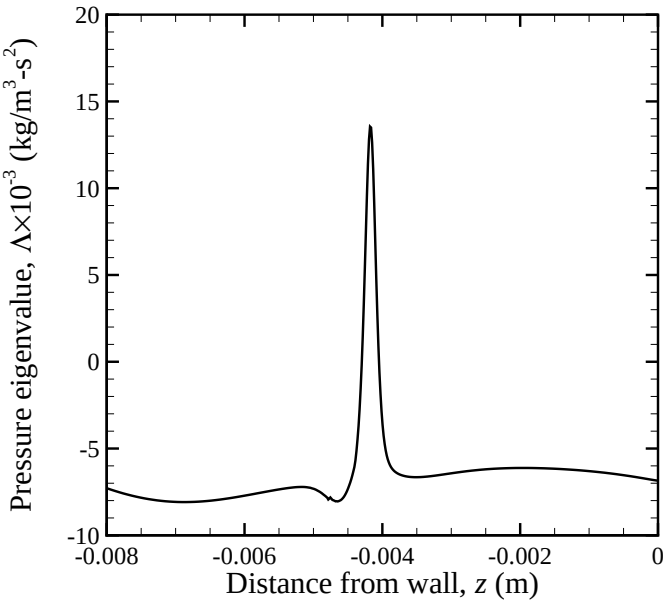


Figure 4.34: The pressure eigenvalue profile for $\Phi = 0.70$ (Case 11). The profile contains a sharp peak at the flame front.

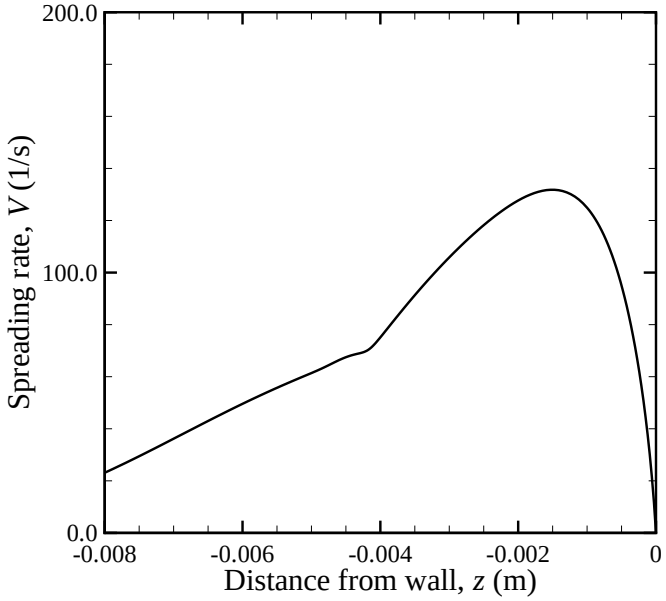


Figure 4.35: The spreading rate profile for $\Phi = 0.70$ (Case 11). A near plateau can be observed between $z = -0.0045$ and $z = -0.004$.

and compared to estimate modeling errors in the GRI-Mech 3.0 chemical kinetics model. Shown in Fig. 4.42 is the range of error at three different equivalence ratios when compared to experimental data reported in Bergthorson (2005). The model works well near the

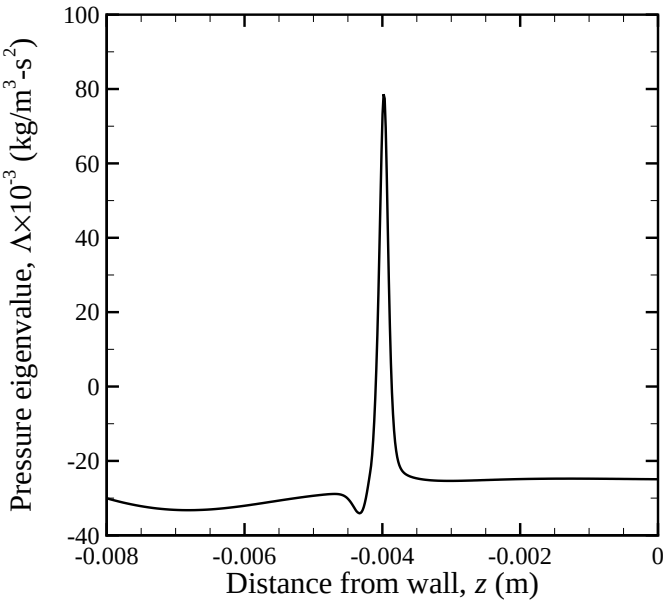


Figure 4.36: Pressure-eigenvalue profile for $\Phi = 0.90$ (Case 3). The peak of the profile is larger compared to that for a weaker flame (cf. Fig. 4.34).

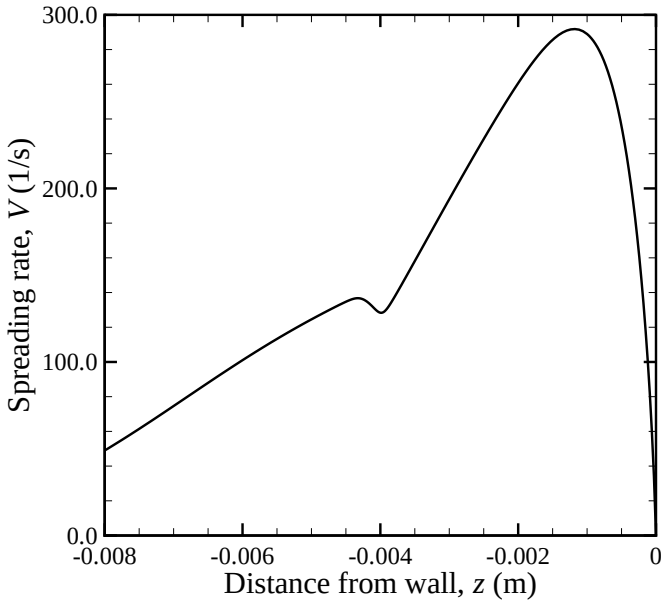


Figure 4.37: Spreading rate for $\Phi = 0.90$ (Case 3). The dip of the profile is the signature of negative flame curvature.

stoichiometric conditions while exhibiting as much as 10% error in lean and rich limits. This result is slightly different from a conclusion drawn in Bergthorson *et al.* (2005b) who relied on one-dimensional simulations. When experimental data are compared to the one-

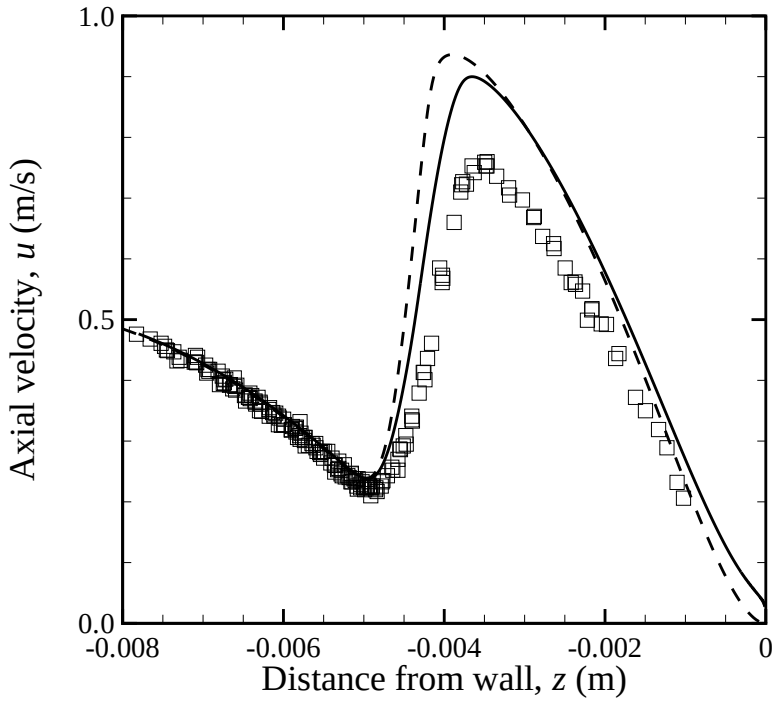


Figure 4.38: Axial velocity profile comparison between simulation and measurements for $\Phi = 0.70$. Squares denote experimental data (Run 212) from Bergthorson (2005), dashed lines are the results of a two-dimensional simulation (Case C070-11), while the solid line is a simulated PSV-measured velocity profile.

dimensional solutions, the results agree well except at lean conditions. On the other hand, intuitively, GRI-Mech may work better on the lean side because it contains a smaller subset of C-2 and C-3 compounds, and reactions involving them. The present work indicates GRI-Mech needs to be improved in a rich mixture case while it also indicates that the mechanism requires more work on the lean side; the latter is the same conclusion reached in Bergthorson *et al.* (2005b).

Turányi *et al.* (2002) found that the uncertainty in flame velocity is typically 2–5 cm/s, considering uncertainty in both thermodynamic and kinetic parameters, but caused mainly by rate constants. Considering less than 10% error at $\Phi = 0.70$ is at the lowest range in the uncertainty, this level of discrepancy between simulation and experiments is reasonable. To obtain further accuracy, as recommended by Turányi *et al.* (2002), rate constants for several important key reactions and thermodynamic data of certain radicals such as hydroxyl radicals need to be examined.

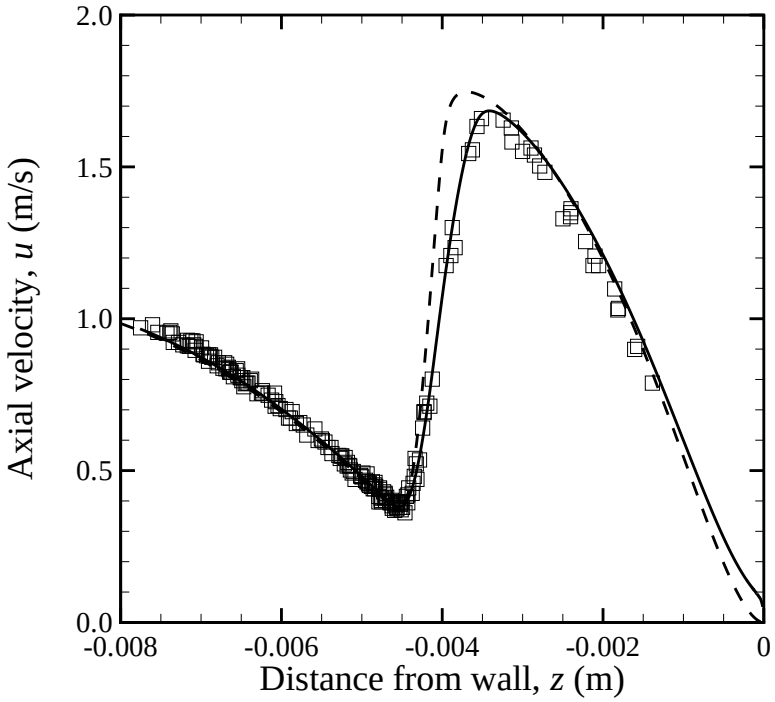


Figure 4.39: The axial velocity profile comparison between simulation (Case C090-4) and measurements (Run 206) from Bergthorson (2005) for $\Phi = 0.90$. (Legend as in Fig. 4.38

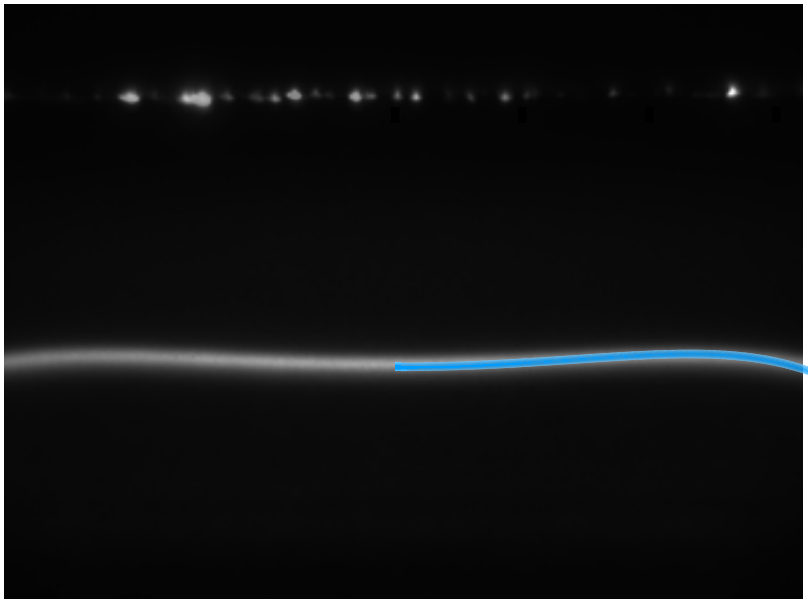


Figure 4.40: The experimental CH-PLIF image from Run 206 ($\Phi = 0.90$), reported in Bergthorson (2005), with computed CH contours (blue) superimposed on the right half. The experimental image is an average of 1,000 instantaneous exposures.

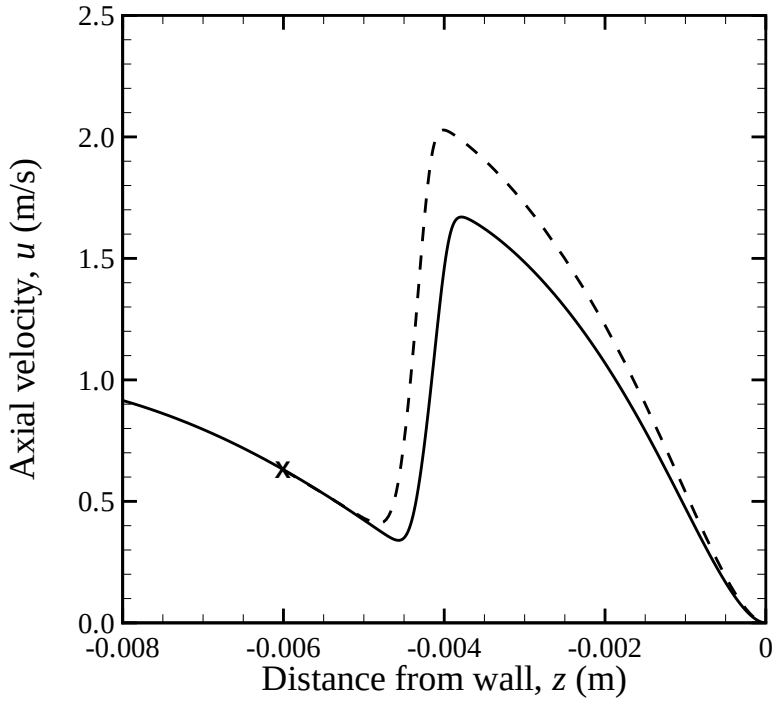


Figure 4.41: The axial velocity profile comparison between simulation (solid line, Case C120-1) and one-dimensional simulation (dashed line). The latter is known to track experimental data well for $\Phi = 1.20$ (Bergthorson, 2005).

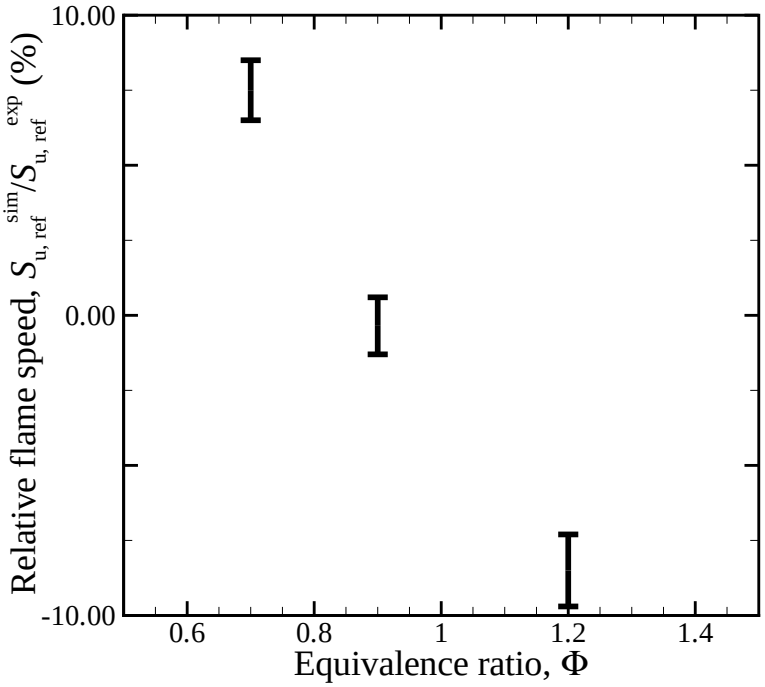


Figure 4.42: Variance between simulated reference flame speed and measured reference flame speed. The model works well near stoichiometric conditions, but there is a variance as high as 10% for rich and lean flames. This result is slightly different from a conclusion drawn in Bergthorson *et al.* (2005b) in which GRI-Mech agreed well with one-dimensional results. The agreement between experimental data and one-dimensional simulation data can be seen in Fig. 4.56 below.

4.2.2.3 Flame speed modification due to external conditions

In the last two sections, the validity of the one-dimensional model and the GRI-Mech 3.0 chemical kinetics model are assessed. The purpose of the current and following sections are to study the behavior of the flame itself numerically. The laminar flame speed, S_u^0, is an important parameter in laminar flame theory, yet a direct measurement of such a quantity is almost impossible. The laminar flame speed is defined as the terminal velocity of a one-dimensional planar, adiabatic flame, propagating into a quiescent unburnt, premixed, combustible mixture. In numerical computation, this quantity can be obtained as the solution to an eigenvalue problem for a premixed flame. However, due to the difficulty of making a direct measurement, many estimates rely on other measurements. The flame burning velocity varies because of various factors such as strain, stretch, curvature of the flame front, heat loss and so on. For the case of stagnation-stabilized flames, Egolfopoulos *et al.* (1997) studied the effect of the stagnation wall and found that it is unimportant as long as the flame is not close to the wall, so that the heat loss to the wall is insignificant. Those simulations relied on the EDL92 (Egolfopoulos *et al.*, 1992) and GRI-Mech 1.2 mechanisms. The discussion following confirms these findings, based on the GRI-Mech 3.0 model, which is used throughout the present study.

Fig. 4.43 shows the effect of the stagnation wall temperature at $\Phi = 0.70$. As can be seen: as the wall temperature rises, the rolling-off part of the curve straightens up, leading to an almost linear profile through a wide range of strain-rates. As pointed out by Egolfopoulos *et al.* (1997), at low enough strain-rates and with flames stabilized far enough from the wall, the impact of wall temperature on flame behavior is minor. However, the question remains: how far is far enough? Shown in Fig. 4.43 is reference flame speed against flame location. As can be seen, the flame propagates faster as wall temperature rises. Wall temperature does affect flame location. As wall temperature rises, the flame burning velocity increases, but at the same time, it projects itself on the characteristic strain-flame speed correlation by finding a right spot on that curve. Another thing to add to the earlier study is that even after the wall temperature is raised, the extrapolation of the strain-flame speed curve does not lead to the laminar flame speed, S_u^0, as can be seen in Fig. 4.43. An earlier study, Egolfopoulos *et al.* (1997) proposed:

> Therefore, it is proposed that the single jet-wall configuration can be used as an

alternative technique for the S_u^0 determination. However, the caution is required to assure that the flame is not affected by the wall as indicated by the "bending" of the $S_{u,ref}$ profile, and that data over a wide range of K[†] are obtained as close to $K = 0$ as possible in order to assume a more meaningful extrapolation.

However, it does not seem the laminar flame speed is on the linear extrapolation of reference velocities when $S_{u,ref}$ is plotted against strain-rate. When $S_{u,ref}$ is plotted against flame stretch instead of strain-rate, this conclusion makes sense, as shown in Fig. 4.45. In the ideal laminar flame speed calculation using a freely propagating flame model, both strain and stretch are zero due to the one-dimensional nature of the model. Shown in Figs. 4.46 through 4.49 are the reference flame speed, $S_{u,ref}$ from Cantera, plotted against the flame stretch, $\Gamma = \sigma_t(x_F)$. Note that these results do not include flame curvature. When the flame is curved, a modification is necessary, as discussed in the next section.

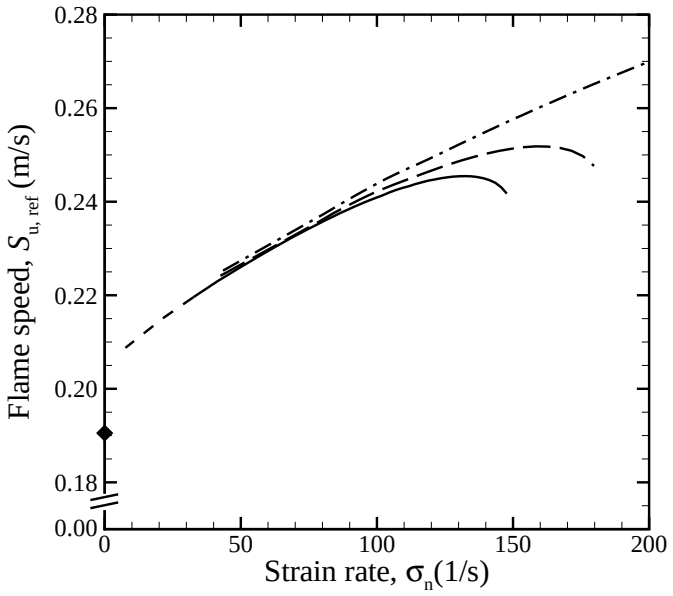


Figure 4.43: The effect of the stagnation wall temperature ($\Phi = 0.70$) studied by one-dimensional model with GRI-Mech 3.0 kinetic model at three different wall temperatures: $T_{wall} = 300\,K$ (solid line), $T_{wall} = 1000\,K$ (long dashed line), and $T_{wall} = 1700\,K$ (dot-dashed line). As T_{wall} approaches the adiabatic flame temperature, the relationship between strain and the flame speed is closer to linear. Short dash indicates cases where the flame is too close to the nozzle boundary condition and the flame data are affected by the nozzle temperature, which is set to 300 K.

[†] K in Egolfopoulos *et al.* (1997) denotes the strain-rate, σ, in this study

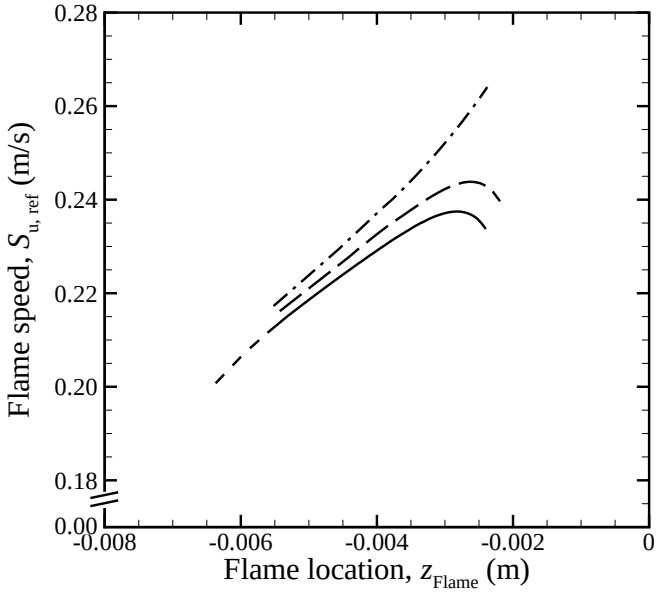


Figure 4.44: Although it does not appear on a σ - $S_{u,ref}$ plot, wall temperature influences where the flame stabilizes itself. When the flame is far enough from the wall, it does not affect the σ - $S_{u,ref}$ relationship. Therefore it is not possible to make a general statement as to what distance from the wall is far enough. (Legend as in Fig. 4.43)

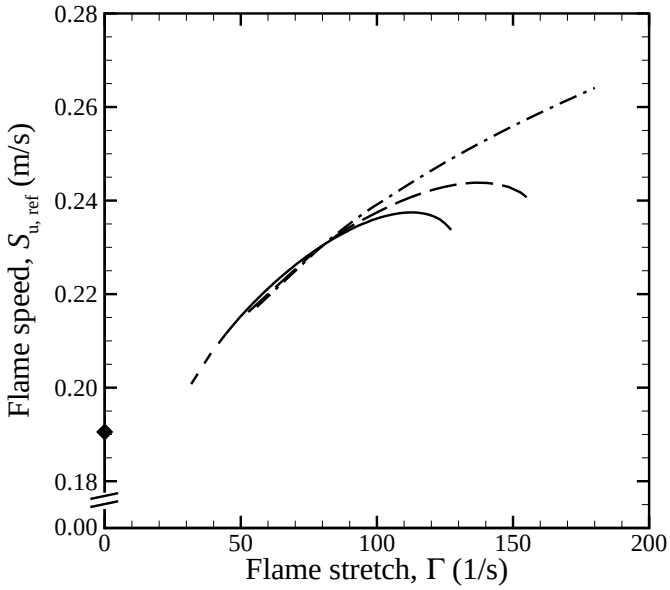


Figure 4.45: When the observed reference flame speed is plotted against flame stretch, it may be possible to estimate the laminar flame speed, S_u^0, by the extrapolation of the data to zero flame stretch at high wall temperature. (Legend as in Fig. 4.43)

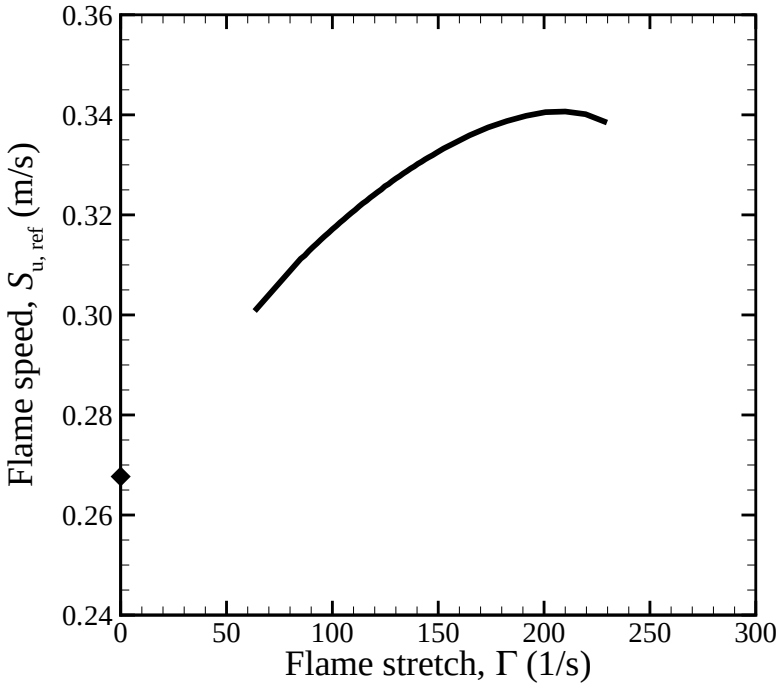


Figure 4.46: Reference flame velocity computed with the one-dimensional model for $\Phi = 0.80$ (solid line) and laminar flame speed, S_u^0 (♦)

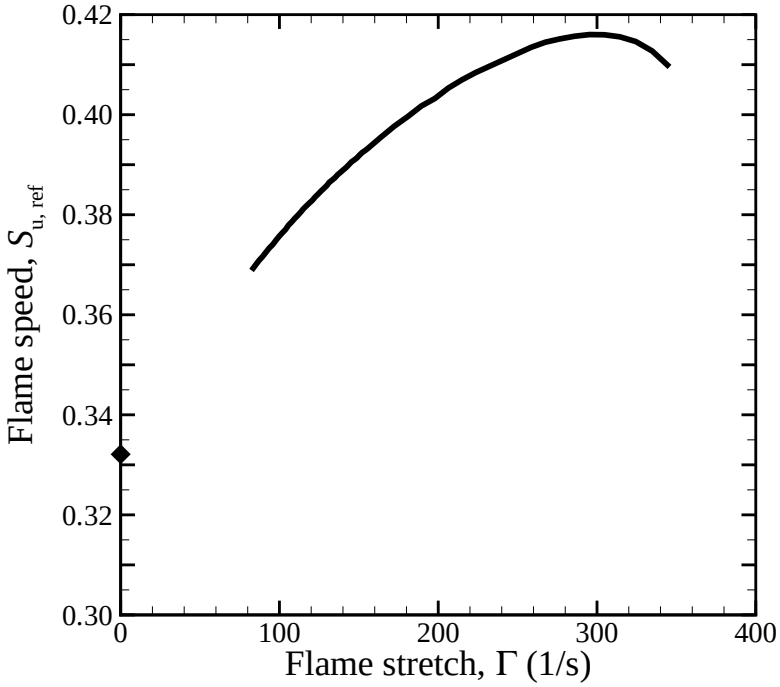


Figure 4.47: For $\Phi = 0.90$. (Legend as in Fig. 4.46)

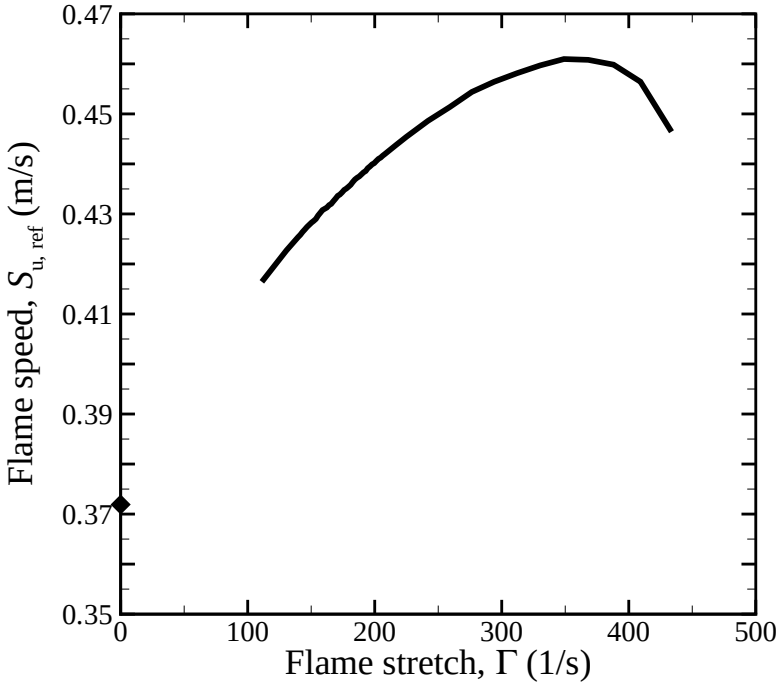


Figure 4.48: For $\Phi = 1.00$. (Legend as in Fig. 4.46)

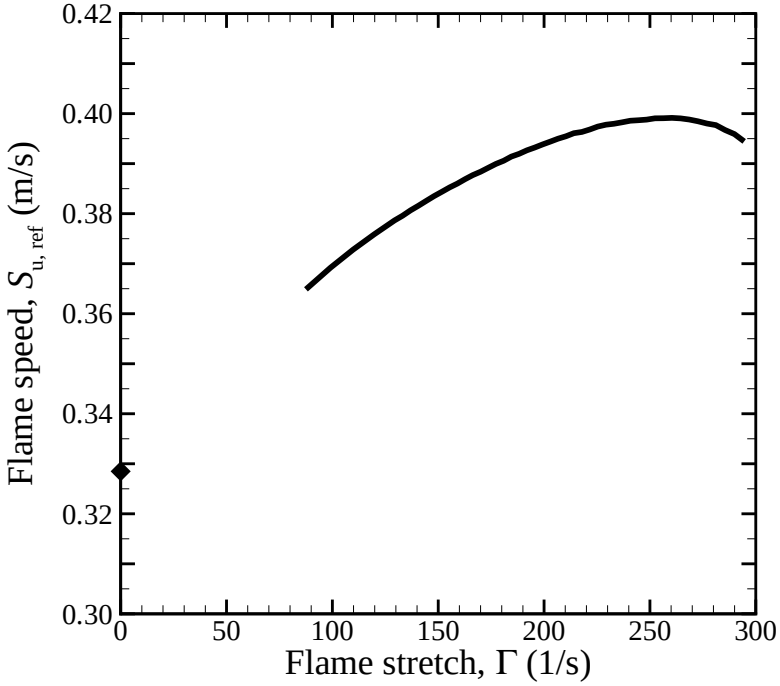


Figure 4.49: For $\Phi = 1.20$. (Legend as in Fig. 4.46)

4.2.2.4 Aerodynamic and geometric effects on flame speed

Interesting properties of a flamelet or a laminar flame are its propagation velocity and its variation with strain-rate (Law & Sung, 2000; Williams, 2000; Bergthorson, 2005), flame stretch (Wu & Law, 1984; Law, 1988; Vagelopoulos & Egolfopoulos, 1998), and curvature (Markstein, 1951; Matalon & Matkowsky, 1982). Flame propagation depends on various aerodynamic, thermodynamic, and chemical-kinetic conditions. However, these conditions are often interrelated, making it difficult to understand flame response to one condition alone. Both two-dimensional simulations and laboratory flames contain finite-curvature effects, whereas one-dimensional models do not.

Shown in Fig. 4.50 is a comparison of reference flame speed as a function of strain-rate for an equivalence ratio of 0.70. The reference flame speed is the local minimum velocity normal to the flame in the unburnt region, whereas the strain-rate is measured by the upstream absolute maximum velocity gradient, using standard definitions in the Community. The results from two-dimensional simulations and the one-dimensional model show the same trend: flame speed increases as strain-rates increase. The one-dimensional model predicts a somewhat concave shape, which is because of wall heat loss that is consistent with previous work by Egolfopoulos *et al.* (1997). The two-dimensional simulation data are clearly separated by the one-dimensional data. On one side, flames have a positive curvature (denoted as P), meaning they are concave towards the unburnt gas, while on the other side, flames have negative curvature (denoted as N). In this case, positive curvature leads to an increase in flame speed compared to a comparable one-dimensional case that has the same hydrodynamic strain. This modification of flame speed may be accounted for in terms of the phenomelogical model by Markstein (1951).

Landau (1944) and Darrieus, in analyzing the stability of plane flames, assumed that the behavior of wrinkled flames was not affected by the structure of the flame itself, leading to a flame that propagates at a uniform speed. This assumption led to their conclusion that plane flames are unconditionally unstable, which contradicts laboratory observations (see Matalon & Matkowsky, 1982; Clavin & Joulin, 1983). Markstein (1951) proposed that a local instantaneous burning velocity should depend on local instantaneous curvature of the

flame front only, which led to the phenomenological expression for the burning velocity,

$$\frac{S_n}{S_{\mathrm{u}}^0} = 1 + \mathcal{L}\kappa, \tag{4.8}$$

where S_n is the flame-normal propagation velocity, S_{u}^0 is the laminar flame speed at zero stretch and zero curvature, κ is the local curvature of the flame, and $\mathcal{L}$ is the Markstein length that is considered to be a characteristic length of the order of the flame thickness. Sometimes the Markstein length is scaled by the diffusion flame thickness to obtain the Markstein number, $Ma = \mathcal{L}/\delta_{\mathrm{flame}}$. In later work, Markstein (1964) recognized that it was the curvature of flame front relative to the curvature of the flow field that affects burning velocity and derived a more complete expression, which can be written as (following Clavin & Joulin (1983)):

$$\frac{S_{\mathrm{u,ref}}(\kappa)}{S_{\mathrm{u}}^0} = 1 + \mathcal{L}\left(\kappa + \mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{n}/S_{\mathrm{u}}^0\right), \tag{4.9}$$

where the first term is the effect of curvature, and the second term is the hydrodynamic stretch, which is related to the local stretch of the flame, Γ:

$$\Gamma \equiv \frac{1}{A}\frac{dA}{dt} = -\mathbf{nn} : \nabla\mathbf{u} + \nabla\cdot\mathbf{u} + S_L\nabla\cdot\mathbf{n}, \tag{4.10}$$

where $\mathbf{n}$ is the normal vector of the flame front pointing upstream, and S_L is the flame propagation velocity (which is zero in the case of stationary flames). Candel & Poinsot (1990) derived the above decomposition for flame stretch and noted that flame stretch has three contributions: the first term is because of the strain-rate, the second term is the volumetric expansion of the fluid, and the last term is because of the flame curvature. However, in the case of a stationary flame, the above expression becomes,

$$\Gamma = \nabla_t\cdot\mathbf{u}_t, \tag{4.11}$$

i.e., the spreading rate evaluated at the flame front. The operator, ∇_t, is the flame tangential component of the ∇ operator (Chung & Law, 1984). The effect of curvature is implicitly included in Γ through the change in tangential velocity, whereas it was explicit for the non-steady flame as in Eqn.(4.10).

Matalon & Matkowsky (1982) and Clavin & Joulin (1983) derived the above expression

for flame speed using high activation energy asymptotics, assuming small strain. They clarified that there are two contributions to total stretch: one from curvature and one from hydrodynamic strain (non-uniformity of the underlying flow). However, as noted by Tien & Matalon (1991), flame speed depends on where the reference point is taken to measure it: for example, the location of the minimum velocity upstream of the flame (Wu & Law, 1984; Egolfopoulos *et al.*, 1997; Bergthorson *et al.*, 2005b), that of the velocity maximum (Mendes-Lopes & Daneshyar, 1985), or the linearly or quadratically extrapolated velocity at the maximum heat release (Davis *et al.*, 2001; Deshaies & Cambray, 1990). Tien & Matalon (1991) investigated the implications of choosing three different reference points and found that when the downstream edge of the preheat zone is used as the reference location, flame speed indeed decreases with increased stretch, whereas when the velocity minimum is used, flame speed increases with increased stretch. In the present study, the location of minimum velocity upstream of the flame is used. The flame-speed measure at this location is different from the one used in the above analysis, and care must be taken to interpret the data consistently with this difference. Furthermore, previous work assumed only a small perturbation from a planar flame, and the correction term is on the order of $\epsilon = \delta_{\text{flame}}/L$, where δ_{flame} is the flame thickness and L is the length scale of the problem; these are typically on the order of $10^{-4}\,\text{m}$ and $10^{-2}\,\text{m}$ for the cases considered here, respectively.

Since the curvature effect can be accounted for by Markstein's model, Eqn. (4.8), we correct the curvature effect on flame speed through the following equation:

$$S_{\text{u,ref}}^{\kappa=0}(\sigma) = \frac{S_{\text{u,ref}}(\sigma,\kappa)}{1+\mathcal{L}\kappa} \tag{4.12}$$

to obtain flame speeds that should be observed in the absence of flame front curvature. This equation implies a slight modification to the originally proposed relation. The original equation as well as its extension by Matalon & Matkowsky (1982) and Clavin & Joulin (1983) assumed a planar freely propagating flame, and its perturbation from such an ideal state is described by the equations proposed by them. Since these models are for flows with small strain, the idea here is to take a curvature-free strain-stabilized flame as a reference state instead of planar freely propagating flame as was done in Markstein's original work. Curvature effects should still appear in the same manner at leading order. Bradley

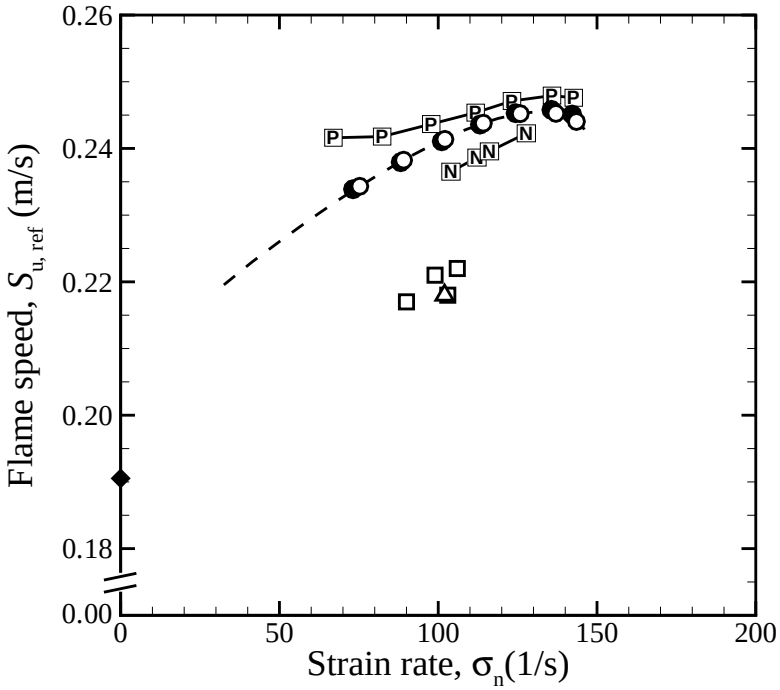


Figure 4.50: Comparison of calculated flame speed as a function of strain-rate, at an equivalence ratio of 0.70. (P): Two-dimensional simulations (Phase II, $\kappa > 0$); (N): Two-dimensional simulations (Phase III, $\kappa < 0$), (dashed line): One-dimensional simulations with plug-flow BC; (□): measurements by Bergthorson (2005); (△): measurements by Benezech *et al.* (2006), (⧫): Laminar flame speed computed using a one-dimensional freely propagating flame model. Circles indicate the cases corresponding to Phase II simulations, with the white ones indicating the plug-flow BC (1D) while the black ones indicate the slope-matched BC (1D-s).

et al. (1996) conducted a numerical simulation of one-dimensional spherically propagating methane flames using a reduced mechanism to estimate the Markstein number, Ma, for several different mixtures. They have found that Markstein numbers for curvature and stretch are different. Therefore, instead of a single number, Ma, there are Ma^c for curvature and Ma^s for stretch term. Using their Markstein number for curvature, Ma^c, of methane at $\Phi = 0.70$, 1.47, the right-hand side of the above equation can be calculated. This can remove the curvature effect from the observed reference flame speed, $S_{\mathrm{u,ref}}$. The result is plotted in Fig. 4.51. This correction method seems to work well and can organize differently curved flames into a single-variable description, permitting flame behavior to be discussed in terms of strain-rate only. It should be pointed out that it should be possible to apply the same method to experimental data. Curvature of the flame front can be estimated by

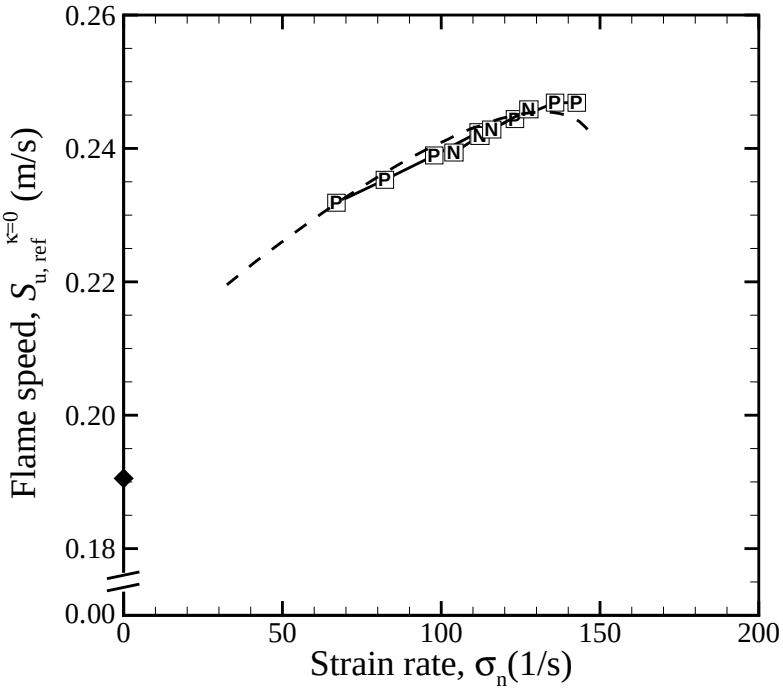


Figure 4.51: Curvature effect is corrected through Markstein's model using the previously reported Markstein number of 1.47 for $\Phi = 0.70$. Legends as in Fig. 4.50.

fitting a circle or polynomial to the CH-PLIF image, shown in Fig. 4.40, for example.

As noted earlier, it may be useful to study flame behavior in terms of flame stretch, instead of the strain-rate of the flow field. This is particularly true if a connection between flame speed obtained from strain-stabilized flame and freely-propagating flame is considered. However, when the flame is curved, attention must be paid to the appropriate stretch that is used to organize the data. To better understand the behavior of flames under stretch, we decompose stretch into two components, similar to the decomposition of flame stretch by Candel & Poinsot (1990), but for the steady case:

$$\Gamma = \Gamma_{1D} + \Gamma_c \tag{4.13}$$

where Γ_{1D} is a contribution from strain-rates (non-uniformity of the flow) and dilatation, while Γ_c is from curvature. Fig. 4.52 plots flame stretch vs. strain-rate for an equivalence ratio of 0.70. It can be seen that Γ_c is positive for positively curved flames and Γ_c is negative for negatively curved flame, provided $\Gamma_c = 0$ in the one-dimensional model. If curvature effects can be corrected through Markstein's model, $\Gamma_{1D} = \Gamma - \Gamma_c$ must be used to organize

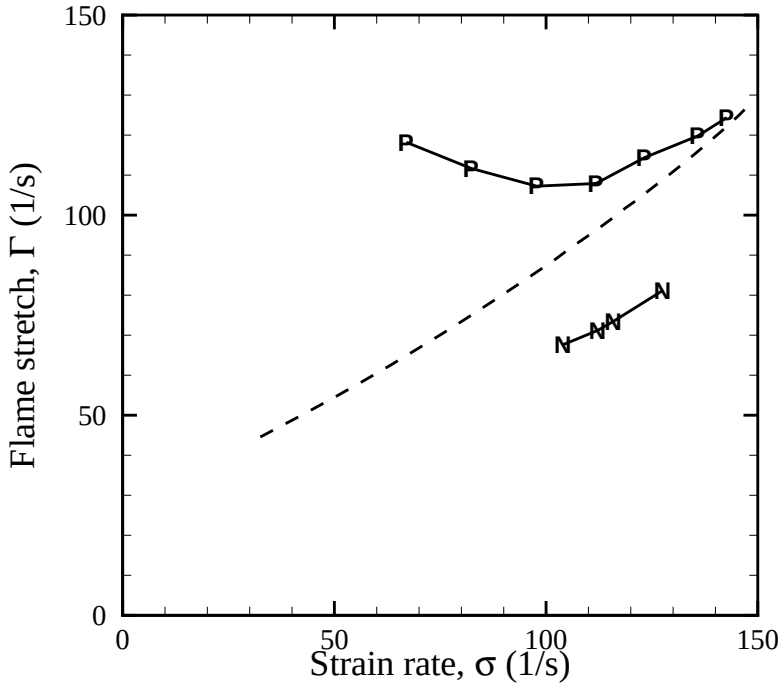


Figure 4.52: Flame stretch (Γ) as a function of the strain-rate (σ) for $\Phi = 0.70$. The two-dimensional simulation data with positive curvature (P) asymptote to the results from the one-dimensional model, as flow rate (and strain rate) increases. Since the one-dimensional model contains no curvature, the contribution of stretch below the curve of the one-dimensional model data is from dilatation, while the contribution above it is from the geometric curvature effect.

flame speed data, instead of the total stretch, Γ. Then, the following equation

$$S_{\mathrm{u,ref}}^{\kappa=0}(\Gamma_{\mathrm{1D}}) = \frac{S_{\mathrm{u,ref}}(\kappa,\Gamma)}{1+\mathcal{L}\kappa} \tag{4.14}$$

can be extrapolated so that $S_{\mathrm{u,ref}}^{\kappa=0}(0) = S_{\mathrm{u}}^{0}$. This is necessary because, although increasing curvature and increasing aerodynamic strain rate both lead to increased flame stretch and increased flame speed, they contribute at a different rate, unlike the asymptotic model that Eqn.(4.9) predicts.

However, as equivalence ratio increases, flame speed obtained from the one-dimensional simulations deviates from a region bounded by flame speeds of positively curved and negatively curved flames of the two-dimensional simulations, although positively curved flames still propagate faster than negatively curved flames. Shown in Figs. 4.53–4.56 are the results from $\Phi = 0.80$, 0.90, 1.00, and 1.20, respectively. Results for $\Phi = 0.90$ are scaled by an

appropriate Markstein number taken from Bradley *et al.* (1996), as was done for the case of $\Phi = 0.70$, and shown in Fig. 4.57. This departure of one-dimensional from two-dimensional ones results appears to be caused by the finite extension of the flame in the two-dimensional simulation, which supports diffusion of species in the radial direction, another effect that is absent in one-dimensional models. This effect may be quantified by additional simulations with larger flame radius to assess effects of the flame size in radial direction.

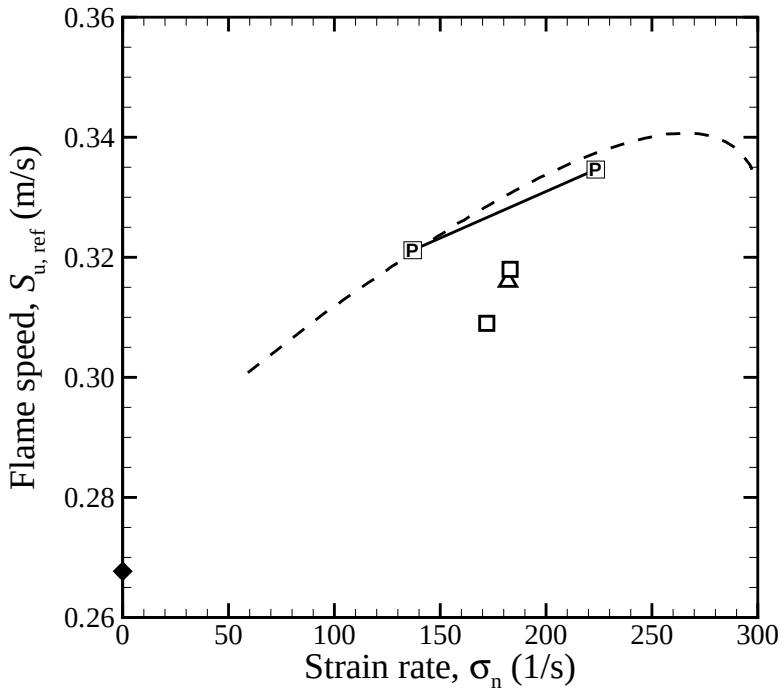


Figure 4.53: Comparison of calculated flame speed as a function of strain-rate, for an equivalence ratio of 0.80. (Legend as in Fig. 4.50.)

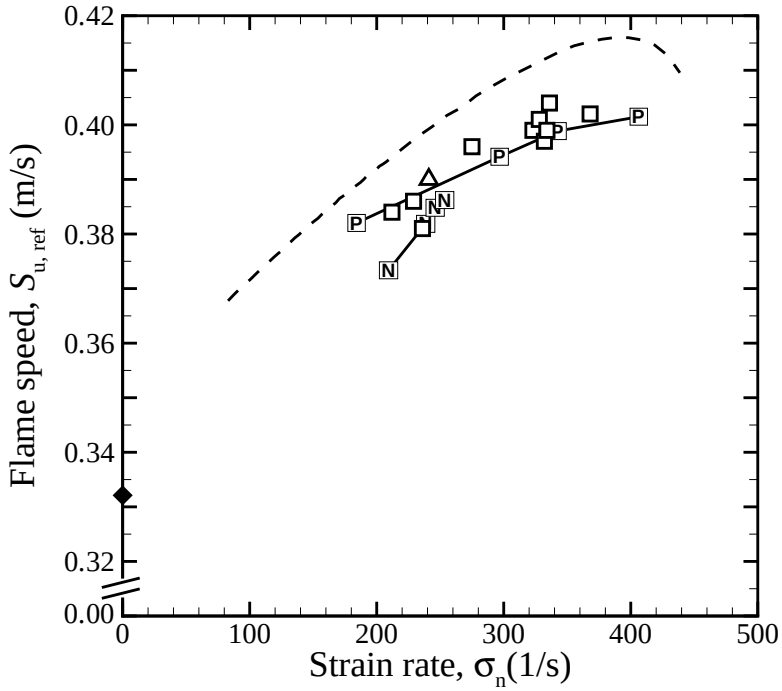


Figure 4.54: Comparison of calculated flame speed as a function of strain-rate, for an equivalence ratio of 0.90. (Legend as in Fig. 4.50.)

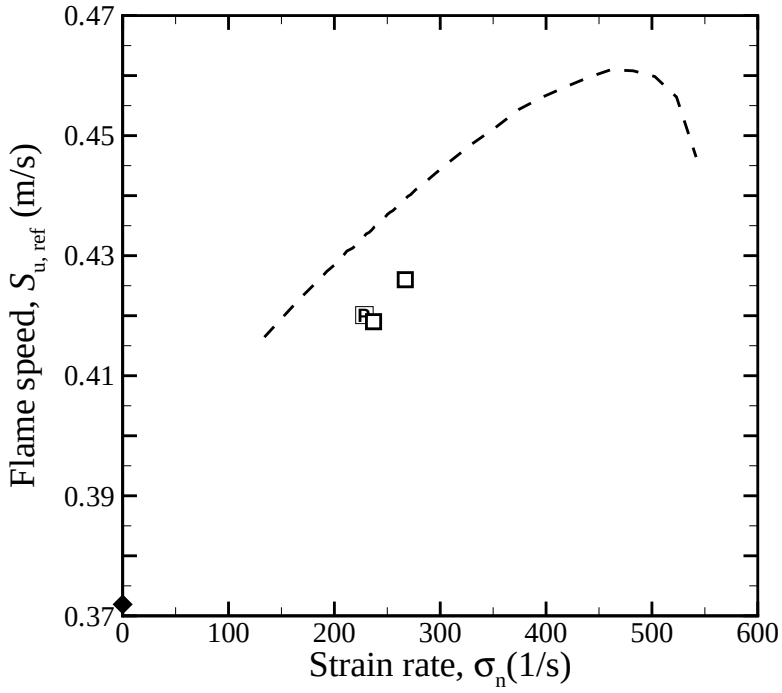


Figure 4.55: Comparison of calculated flame speed as a function of strain-rate, for an equivalence ratio of 1.00. (Legend as in Fig. 4.50.)

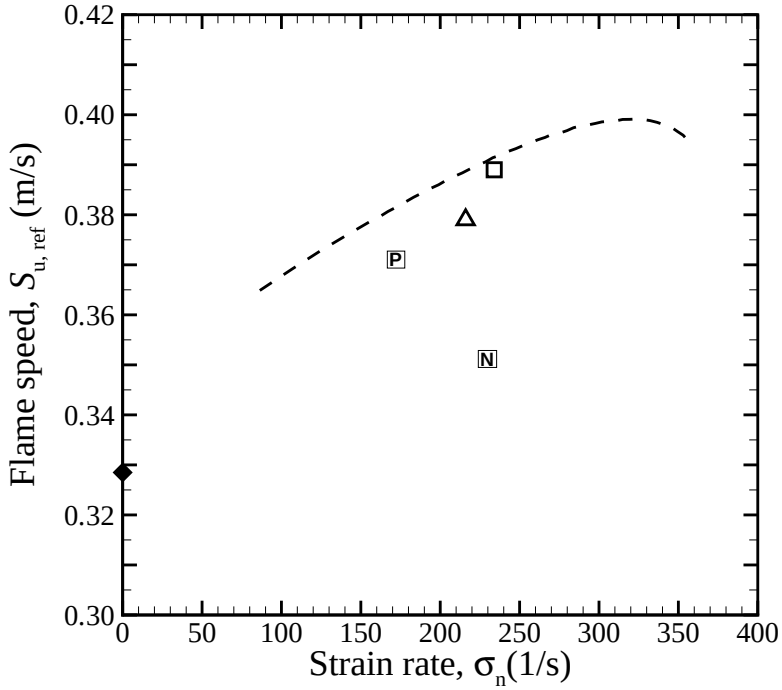


Figure 4.56: Comparison of calculated flame speed as a function of strain-rate, for an equivalence ratio of 1.20. (Legend as in Fig. 4.50.)

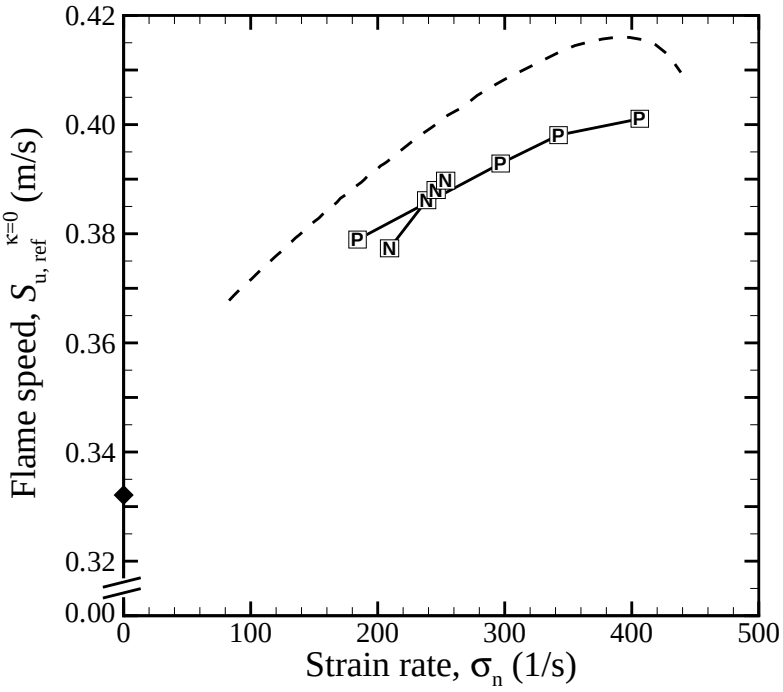


Figure 4.57: Curvature effect is corrected through Markstein's model with the previously reported Markstein number of 2.20 (Bradley *et al.*, 1996), for $\Phi = 0.90$. (Legends as in Fig. 4.50)

4.2.3 Summary

A comparison of experimental data and axisymmetric, two-dimensional simulations of the corresponding laboratory phenomena revealed that the methane combustion model based on GRI-Mech 3.0 works well for methane-air mixtures near stoichiometry. However, GRI-Mech 3.0 leads to an overprediction of laminar flame speed for lean mixtures, and an underprediction for rich mixtures. The latter is at variance with previous predictions based on one-dimensional models. This discrepancy stems from the fact that the one-dimensional model lacks the lateral diffusion of important species arising from the finite radius of curvature of the flames measured in the laboratory, and as simulated. The geometry of the flame front—such as the finite radius and finite curvature—is shown to have a significant contribution to flame propagation speed. In particular, Markstein's theory on the flame speed modification for flame curvature is confirmed by the present study as necessary for steady stagnation flames. At all equivalence ratios studied, Markstein length and Markstein numbers are always positive for methane-air mixtures, *i.e.*, positive curvature leads to faster flame propagation. There are some experimental (Deshaies & Cambray, 1990) and numerical (Davis *et al.*, 2002a,b) studies on the Markstein length, using stagnation flame or opposed-jet flames. In these reports, different definitions of flame speed and stretch are used from the ones used in the present study, and it more analysis is required to conclude whether the curvature effect that appears in the present study can be explained by the Markstein model and Markstein numbers previously reported. Currently, there is no consensus as to what the Markstein number is, and more experimental and numerical work, including two-dimensional simulation using full chemistry, appear necessary to settle such issues. Once settled, such information may be used to adjust experimentally measured flame speeds so that the target data for chemical kinetics can be organized without contamination by curvature effects.

A one-dimensional model with plug-flow boundary condition does not yield satisfactory results when a prediction of flame location is important. However, it is useful in providing a flame-speed to strain-rate relationship because the velocity boundary condition has an insignificant impact on flame location relative to local flow conditions. For this purpose, the plug-flow boundary conditions may be used. However, its comparison with experimental data requires some caution, because of effects not accommodated in the one-dimensional

model, such as the finite radial extension and the finite curvature of the flame.

Chapter 5

Conclusion

In the end, a theory is accepted not because it is confirmed by conventional empirical tests, but because researchers persuade one another that the theory is correct and relevant.—Fischer Black

5.1 Concluding remarks

A new algorithm for unsteady chemically reacting flow has been developed and implemented. The code has undergone verification tests to ensure the correctness of the obtained numerical results. In the process, a new basis function set for use in the axisymmetric spectral element method has been developed that incorporates the proper parity condition near the axis. This new basis function is an extension of early works by Leonard & Wray (1982) and Matsushima & Marcus (1995) for spectral methods. The resulting computing framework was used to study the behavior of non-reacting impinging jets and chemically reacting premixed methane flames in stagnation flow. The spectral element method was proved useful in the simulation of laboratory-scale flames, since it can accommodate the details of the experimental setup without sacrificing accuracy. Its extension to large density variations and reacting flow also proved successful.

For cold flow, a one-parameter model that describes the centerline velocity profile of the impinging jet is obtained, in collaboration with other researchers via synergetic efforts between experiments and computations (Bergthorson *et al.*, 2005a). This model can be used to specify an appropriate boundary condition for the one-dimensional model. In particular, the simulation played a key role in obtaining a new scaling law in terms of an effective nozzle diameter.

In the case of reacting flow, variances arising from the use of the GRI-Mech 3.0 chemical

kinetic model were quantified, and it was shown to work well near stoichiometric conditions. This comparison was made possible by accomodating every important factor in the experimental setup and is the first direct comparison of laboratory measurements and numerical simulations on the stagnation flames. The result of the comparison shows that the GRI-Mech 3.0 chemical kinetics model could be improved, at least for lean and rich methane mixtures. The conclusion for rich mixtures is new and differs from previous estimates that were obtained by comparing experiments and one-dimensional numerical solutions. This discrepancy is most probably caused by the fact that the one-dimensional model lacks the radial diffusion of important species due to its assumption of a flat (zero-curvature) flame of infinite extent. This effect does not seem to have been explicitly noted in previous studies that relied on one-dimensional hydrodynamics models to study stagnation flames, and more study is necessary to confirm and quantify the effect.

Flame-front curvature was shown to have a discernible effect on the resulting flame speed, and correction is necessary to have an accurate estimate of errors in chemical kinetics models in the future, or to obtain target flame-speed data from experiments used for kinetic rate parameter optimization. Although not recommended, if the extrapolation method is used to estimate laminar flame speed, this curvature correction is important as flames can have different curvatures for each single realization. Of course it would be more accurate to make comparisons and set benchmarking target for mechanism optimizations without flame speed extrapolations, as implemented by Bergthorson *et al.* (2005b). The fact that the estimation of Markstein numbers remains an on-going research problem may cause difficulty in adjusting flame speed to take curvature effects into account.

Using an axisymmetric two-dimensional simulation environment, it was shown that one-dimensional models can provide a flame-speed to strain-rate relationship because the velocity boundary condition has an insignificant impact on the flame speed itself although the prediction of flame location *is* sensitive to a choice of velocity boundary conditions.

Many results reported here have benefitted from the concurrent experimental and numerical work, conducted by a team, and such an approach was proven to be powerful and beneficial in understanding flame behavior, although it was not particularly easy to ensure that both the experimental and numerical details were in sufficient agreement to permit the quantitative comparisons performed.

5.2 Recommended future work

As noted in the first chapter, each methane combustion model includes a hydrogen reaction submechanism. Although much simpler, numerical results for hydrogen flames do not agree with each other (see Appendix E), implying slightly different behavior for each hydrogen submechanism of the hydrocarbon reaction mechanisms. However, one of the most important reactions in determining laminar flame speed of methane is (Williams, 2000),

$$H + O_2 + M \Leftrightarrow HO_2 + M.$$

Turányi *et al.* (2002) also pointed out seven elementary reactions that contribute to uncertainty in methane flame speeds. Two of them are in the hydrogen kinetic submodels: one is the above reaction, and the other is

$$O_2 + H \Leftrightarrow OH + O.$$

It would be useful to quantify errors in the hydrogen reaction subset of each model first, since the hydrogen submechanism is an important ingredient in hydrocarbon mechanisms, before attempting to optimize hydrocarbon kinetic models. Two-dimensional simulation of hydrogen flames are significantly less computationally expensive, and the Omega code can contribute to such developments. In particualr, extinction and reignition behavior of hydrogen flames is expected to be useful in assessing the behavior of chemical kinetics models. The experimental study by Pellet *et al.* (1998) may provide a useful target.

Although in two-dimensional simulations the entire flow field is obtained, the analysis of flame behavior is only discussed along the axis, where the detailed experimental data cited above were obtained. It would be interesting to extend the analysis in the off-axis regions along each streamline. This involves computation of curvature and flame stretch at off-axis locations. It is recommended to start this work for $\Phi = 0.90$ or stoichiometric conditions where experimental and numerical data agree, at least along the axis.

In terms of computing capability, one extension of the code would be to incorporate the multicomponent diffusion model (Curtiss & Bird, 1999; Hirschfelder *et al.*, 1954), including Soret effects in the species diffusion velocity, instead of using the simpler Fickian diffusion model. In particular, Ern & Giovangigli (1998) showed that Soret effects have an impact

on the structure of rich Bunsen flame. EGLIB developed by Ern and Giovangigli, which implements multicomponent transport algorithms, may be useful for this purpose (see Ern & Giovangigli, 1994).

Another useful approach would be to implement a Newton-Krylov method (see Gropp *et al.*, 2000) for solving nonlinear equations numerically. The current time-stepping approach will also be useful within the solution search algorithm as used in CHEMKIN and Cantera. In Chapter 2, it was shown that matrix reuse can save computational time significantly. A similar approach might work for Jacobians of chemistry ODEs. This approach has not been explored and may be worth investigating. Currently, the code makes no attempt in load balancing, and therefore, some processors are taking more elements that are heavily involved with chemical reactions while others are waiting to proceed. The load balancing in terms of the workload in chemistry ODE integration may lead to significant speedup of execution. Finally, the inclusion of a nonconforming mesh and an adaptive mesh refinement algorithm will make the code friendlier to those who do not have a lot of experience with numerical analysis, since allocation and distribution of elements are probably the most difficult task in using the current version of the Omega code, compared to the widely used Cantera program.

Appendix A

Formulation

A.1 The governing equations for compressible reacting flows

The govering equations used for the simulation in this study have been derived before and used extensively (*e.g.*, Williams, 1985; Buckmaster & Ludford, 1982). However, for completeness, we will review those equations and the difficulties they pose to the numerical study.

First the compressible Navier-Stokes equations will be reviewed, and then several different sets of equations under different levels of assumptions will be derived.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{A.1a}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u} - \mathbf{T}) = \rho \mathbf{f}, \tag{A.1b}$$

$$\frac{\partial \rho e_T}{\partial t} + \nabla \cdot (\rho \mathbf{u} e_T - \mathbf{u} \cdot \mathbf{T} + \mathbf{q}) = \rho \mathbf{f} \cdot \mathbf{u} \tag{A.1c}$$

$$\frac{\partial \rho Y_{\mathfrak{m}}}{\partial t} + \nabla \cdot (\rho Y_{\mathfrak{m}} (\mathbf{u} + \mathbf{V}_{\mathfrak{m}})) = \dot{\rho_{\mathfrak{m}}}, \tag{A.1d}$$

where $\mathbf{q}$ is heat flux given by

$$\mathbf{q} = -\lambda_T \nabla T + \rho \sum_{\mathfrak{m}} h_{\mathfrak{m}} Y_{\mathfrak{m}} \mathbf{V}_{\mathfrak{m}} + R_{\mathrm{u}} T \sum_{\mathfrak{m}} \sum_{\mathfrak{n}} \left(\frac{X_{\mathfrak{n}} D_{T,\mathfrak{m}}}{W_{\mathfrak{m}} D_{\mathfrak{m}\mathfrak{n}}} \right) (\mathbf{V}_{\mathfrak{m}} - \mathbf{V}_{\mathfrak{n}}) + \mathbf{q}_R, \tag{A.2}$$

where λ_T is the thermal conductivity, $h_{\mathfrak{m}}$ is the specific enthalpy of species $\mathfrak{m}$, $\mathbf{V}_{\mathfrak{m}}$ is the diffusion velocity for species $\mathfrak{m}$, R_{u} is the universal gas constant, $D_{T,\mathfrak{m}}$ is the thermal diffusion coefficient for species $\mathfrak{m}$, $W_{\mathfrak{m}}$ is molecular weight of species $\mathfrak{m}$, $D_{\mathfrak{m}\mathfrak{n}}$ is binary diffusion

coefficient for species $\mathfrak{m}$ and $\mathfrak{n}$, and $\mathbf{q}_R$ is the radiant heat flux vector.

The stress tensor $\mathbf{T}$ is given by

$$\mathbf{T} = -p\mathbf{I} + \mu\left((\nabla\mathbf{u}) + (\nabla\mathbf{u})^t\right) + \lambda(\nabla\cdot\mathbf{u})\mathbf{I}. \tag{A.3}$$

$\mathbf{V}_\mathfrak{m}$ may be approximated by (Curtiss & Bird, 1999; Bird *et al.*, 2002),

$$\mathbf{V}_\mathfrak{m} = -\frac{D^T_\mathfrak{m}}{T}\nabla T - \sum_\mathfrak{n} D_{\mathfrak{m}\mathfrak{n}}\mathbf{d}_\mathfrak{n} \tag{A.4}$$

where $\mathbf{d}_\mathfrak{n}$ is the diffusion driving force given by,

$$\mathbf{d}_\mathfrak{n} = \nabla X_\mathfrak{n} - (Y_\mathfrak{m} - X_\mathfrak{m})\frac{\nabla p}{p}. \tag{A.5}$$

The first term in Eqn. (A.4) is called Soret effect, which is an effect of thermal diffusion. The species diffusion velocity combines concentration diffusion and pressure diffusion effects, both of which are described in the diffusion driving force.

In these equations, and throughout the thesis, ρ is reserved for density, $\mathbf{u}$ for the velocity vector, e_T for the total energy, $Y_\mathfrak{m}$ for the mass fraction of species $\mathfrak{m}$, and $X_\mathfrak{m}$ for the mole fraction of species $\mathfrak{m}$.

For the transport properties, μ denotes the dynamic viscosity, λ denotes the second coefficient of viscosity, which is equal to $\mu_\mathrm{B} - 2/3\mu$, where μ_B is the bulk viscosity. λ_T denotes the thermal conductivity, $D_{\mathfrak{m}\mathfrak{n}}$ denotes the binary diffusion coefficient between species $\mathfrak{m}$ and $\mathfrak{n}$, and $D_{T,\mathfrak{m}}$ denotes the thermal diffusion coefficient as already defined.

A.1.1 Derivation of other forms of the energy equation

First, we derive several forms of the energy transport equation.

We start with the total energy conservation equation,

$$\frac{\partial\rho e_T}{\partial t} + \nabla\cdot(\rho\mathbf{u}e_T - \mathbf{u}\cdot\mathbf{T} + \mathbf{q}) = \rho\mathbf{f}\cdot\mathbf{u}. \tag{A.6}$$

We then use the momentum equation to obtain the transport equation for the internal energy,

$$\rho\frac{De}{Dt} = \mathbf{T} : \nabla\mathbf{u} - \nabla\cdot\mathbf{q}. \tag{A.7}$$

We then use

$$e = h - p/\rho \tag{A.8}$$

to get the enthalpy transport equation,

$$\rho \frac{Dh}{Dt} = \frac{Dp}{Dt} + \mathbf{\Sigma} : \nabla \mathbf{u} - \nabla \cdot \mathbf{q}, \tag{A.9}$$

where $\mathbf{\Sigma} = 2\mu \mathbf{S} + \lambda(\nabla \cdot \mathbf{u})\mathbf{I}$ is the viscous stress tensor. Eqns. (A.6), (A.7), and (A.9) are equivalent, and there are no additional assumptions in deriving one from another. The enthalpy of the mixture is given by,

$$h = \sum_{\mathfrak{m} \in \mathcal{M}} h_{\mathfrak{m}} Y_{\mathfrak{m}}, \tag{A.10}$$

where,

$$h_{\mathfrak{m}}(T) = h_{\mathfrak{m}}^{0,T_{\text{ref}}} + \int_{T_{\text{ref}}}^{T} C_{p,\mathfrak{m}}(T')dT'. \tag{A.11}$$

Eqn.(A.9) can be written in terms of the temperature evolution as follows:

$$\rho C_{p,\text{mix}} \frac{DT}{Dt} = \frac{Dp}{Dt} + \mathbf{\Sigma} : \nabla \mathbf{u} - \nabla \cdot \mathbf{q} + \sum_{\mathfrak{m}} \nabla \cdot (\rho h_{\mathfrak{m}} Y_{\mathfrak{m}} \mathbf{V}_{\mathfrak{m}}) - \sum_{\mathfrak{m}} \rho Y_{\mathfrak{m}} \mathbf{V}_{\mathfrak{m}} \cdot \nabla h_{\mathfrak{m}} - \sum_{\mathfrak{m}} h_{\mathfrak{m}} \dot{\rho_{\mathfrak{m}}}. \tag{A.12}$$

This form of the energy equation is used later to obtain the low Mach number limit of the energy equation.

A.2 The governing equations in the low Mach number limit

A.2.1 The momentum equation

Following Majda & Sethian (1985) and McMurtry *et al.* (1986), we derive the low Mach number limit of the Navier-Stokes equations with a special consideration to bulk viscosity. First we nondimensionalize the equations:

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot \left(\rho \mathbf{u}\mathbf{u} + p\mathbf{I} - \mu \left[(\nabla \mathbf{u}) + (\nabla \mathbf{u})^t \right] - \lambda (\nabla \cdot \mathbf{u}) \mathbf{I} \right) = \rho \mathbf{f}. \tag{A.13}$$

Here, μ is the dynamic viscosity and λ is the second coefficient of viscosity, which is related to the bulk viscosity by $\mu_{\text{B}} = \lambda + 2/3\mu$.

We scale the primary variables: $\hat{\rho} = \rho/\rho_0$, $\hat{\mathbf{u}} = \mathbf{u}/u_0$, $\hat{x} = x/x_0$, $\hat{t} = t/t_0$ where $t_0 = x_0/u_0$, $\hat{p} = p/p_0$, and $\hat{\mathbf{f}} = \mathbf{f}/({u_0}^2/x_0)$. Furthermore, define ${c_0}^2 = p_0/\rho_0$, $\hat{\mu} = \mu/\mu_0$, $\hat{\lambda} = \lambda/\mu_0$, $M = u_0/c_0$ and $Re = \rho_0 u_0 x_0/\mu_0$. The subscript 0 denotes the reference values of the corresponding variables which are constants. M denotes the Mach number (Ma is researved for the Markstein number in this study.)

$$\frac{\partial \hat{\rho}\hat{\mathbf{u}}}{\partial \hat{t}} + \hat{\nabla}\cdot\left(\hat{\rho}\hat{\mathbf{u}}\hat{\mathbf{u}} + \frac{1}{M^2}\hat{p}\mathbf{I} - \frac{1}{Re}\left(\hat{\mu}\left[(\hat{\nabla}\hat{\mathbf{u}}) + (\hat{\nabla}\hat{\mathbf{u}})^t\right] - \hat{\lambda}(\hat{\nabla}\cdot\hat{\mathbf{u}})\mathbf{I}\right)\right) = \hat{\rho}\hat{\mathbf{f}}. \tag{A.14}$$

When the external force is gravity, *i.e.*, $\mathbf{f} = -g\mathbf{e}_z$, the RHS becomes $-\hat{\rho}/Fr\,\mathbf{e}_z$, where Fr is the Froude number.

Now we perform an asymptotic expansion for small Mach number. Let ϵ be a small parameter:

$$\hat{p}(\hat{x},\hat{t}) = \hat{p}^0(\hat{x},\hat{t}) + \epsilon\hat{p}^1(\hat{x},\hat{t}) + O(\epsilon^2) \tag{A.15a}$$

$$\hat{\mathbf{u}}(\hat{x},\hat{t}) = \hat{\mathbf{u}}^0(\hat{x},\hat{t}) + \epsilon\hat{\mathbf{u}}^1(\hat{x},\hat{t}) + O(\epsilon^2) \tag{A.15b}$$

$$\hat{\rho}(\hat{x},\hat{t}) = \hat{\rho}^0(\hat{x},\hat{t}) + \epsilon\hat{\rho}^1(\hat{x},\hat{t}) + O(\epsilon^2) \tag{A.15c}$$

$$\hat{\mu}(\hat{x},\hat{t}) = \hat{\mu}^0(\hat{x},\hat{t}) + \epsilon\hat{\mu}^1(\hat{x},\hat{t}) + O(\epsilon^2) \tag{A.15d}$$

$$\hat{\lambda}(\hat{x},\hat{t}) = \hat{\lambda}^0(\hat{x},\hat{t}) + \epsilon\hat{\lambda}^1(\hat{x},\hat{t}) + O(\epsilon^2) \tag{A.15e}$$

and substitute these into Eqn.(A.14). We obtain

$$\begin{aligned}&\frac{1}{M^2}\hat{\nabla}\left(\hat{p}^0 + \epsilon\hat{p}^1 + O(\epsilon^2)\right) = \\ &-\frac{\partial \hat{\rho}^0\hat{\mathbf{u}}^0}{\partial \hat{t}} - \hat{\nabla}\cdot\left(\hat{\rho}^0\hat{\mathbf{u}}^0\hat{\mathbf{u}}^0 - \frac{1}{Re}\left(\hat{\mu}^0\left[(\hat{\nabla}\hat{\mathbf{u}}^0) + (\hat{\nabla}\hat{\mathbf{u}}^0)^t\right] - \hat{\lambda}^0(\hat{\nabla}\cdot\hat{\mathbf{u}}^0)\mathbf{I}\right)\right) + \hat{\rho}^0\hat{\mathbf{f}}^0 + O(\epsilon).\end{aligned} \tag{A.16}$$

The leading order of the RHS are the $O(1)$ terms. If $\hat{p}^0$ balances with this, $M \sim 1$ and we simply obtain the original compressible Navier-Stokes equation. If $\hat{p}^1$ balances with the RHS, $M = O(\epsilon^{1/2})$ and $\nabla\hat{p}^0 = 0$ must hold, which dictates that the leading order pressure term is uniform and a function only of time. For external flow, which is the subject of the current study, we assume that the ambient pressure is constant, thus $\hat{p}^0$ is constant and unity for the present study. However, for internal flows, such as reacting flows in an internal

combustion engine, this term can vary with time.

The O(ϵ) equation leads to the low Mach number equations that are used in the present study,

$$\nabla\hat{p}^1 = -\frac{\partial\hat{\rho}^0\hat{\mathbf{u}}^0}{\partial\hat{t}} - \hat{\nabla}\cdot\left(\hat{\rho}^0\hat{\mathbf{u}}^0\hat{\mathbf{u}}^0 - \frac{1}{Re}\left(\hat{\mu}^0\left[(\hat{\nabla}\hat{\mathbf{u}}^0) + (\hat{\nabla}\hat{\mathbf{u}}^0)^t\right] - \hat{\lambda}^0(\hat{\nabla}\cdot\hat{\mathbf{u}}^0)\mathbf{I}\right)\right) + \hat{\rho}^0\hat{\mathbf{f}}^0. \quad \text{(A.17)}$$

This equation is, in dimensional form,

$$\frac{\partial\rho\mathbf{u}}{\partial t} + \nabla\cdot\left(\rho\mathbf{u}\mathbf{u} + (p(x,t) - p_0)\,\mathbf{I} - \mu\left[(\nabla\mathbf{u}) + (\nabla\mathbf{u})^t\right] - \lambda(\nabla\cdot\mathbf{u})\mathbf{I}\right) = \rho\mathbf{f}. \quad \text{(A.18)}$$

Eqn.(A.18) can also be written as follows:

$$\frac{\partial\rho\mathbf{u}}{\partial t} + \nabla\cdot\left(\rho\mathbf{u}\mathbf{u} + p^*\mathbf{I} - \mu\left[(\nabla\mathbf{u}) + (\nabla\mathbf{u})^t\right]\right) = \rho\mathbf{f}, \quad \text{(A.19)}$$

where

$$p^* = (p(x,t) - p_0) - \left(\mu_{\mathrm{B}} - \frac{2}{3}\mu\right)(\nabla\cdot\mathbf{u}). \quad \text{(A.20)}$$

The bulk viscosity is now combined with the purturbational part of the pressure field.

A.2.2 The equation of state

The equation of state is

$$p = \rho\bar{R}_{\mathrm{gas}}T. \quad \text{(A.21)}$$

We normalize this equation as before and we obtain

$$\hat{p} = \hat{\rho}\hat{T} \quad \text{(A.22)}$$

where $\hat{T} = T/T_0$ and $T_0 = p_0/\rho_0\bar{R}_{\mathrm{gas}}$. Again, using the asymptotic expansion, we obtain the equation of state in the low Mach number limit.

$$\hat{p}^0 = \hat{\rho}^0\hat{T}^0. \quad \text{(A.23)}$$

Note that $\hat{T}^0$ is defined as $\hat{T}(\hat{x},\hat{t}) = \hat{T}^0(\hat{x},\hat{t}) + \epsilon\hat{T}^1(\hat{x},\hat{t}) + O(\epsilon^2)$

The LHS of the equation is constant, and thus this equation relates density and temperature, and they cannot vary independently.

Also note that the reference states satisfy the equation of state, $p_0 = \rho_0 \bar{R}_{\text{gas}} T_0$.

A.2.3 The energy equation

Starting from the temperature transport equation,

$$\rho C_{p,\text{mix}} \frac{DT}{Dt} = \frac{Dp}{Dt} + \boldsymbol{\Sigma} : \nabla \mathbf{u} - \nabla \cdot \mathbf{q} + \sum_{\mathfrak{m}} \nabla \cdot (\rho h_{\mathfrak{m}} Y_{\mathfrak{m}} \mathbf{V}_{\mathfrak{m}}) - \sum_{\mathfrak{m}} \rho Y_{\mathfrak{m}} \mathbf{V}_{\mathfrak{m}} \cdot \nabla h_{\mathfrak{m}} - \sum_{\mathfrak{m}} h_{\mathfrak{m}} \dot{\rho_{\mathfrak{m}}}. \tag{A.24}$$

In a similar way to the momentum equations, we obtain the following leading-order equations.

$$\begin{aligned} \hat{\rho} \hat{C}_{p,\text{mix}} \frac{D\hat{T}}{D\hat{t}} = \frac{D\hat{p}}{D\hat{t}} + \frac{M^2}{Re} \hat{\sigma} : (\hat{\nabla} \hat{\mathbf{u}}) - \frac{1}{RePr} \hat{\nabla} \cdot \hat{\mathbf{q}} \\ + \sum_{\mathfrak{m}} \hat{\nabla} \cdot \left(\hat{\rho} \hat{h_{\mathfrak{m}}} Y_{\mathfrak{m}} \hat{\mathbf{V}}_{\mathfrak{m}} \right) - \sum_{\mathfrak{m}} \hat{\rho} Y_{\mathfrak{m}} \hat{\mathbf{V}}_{\mathfrak{m}} \cdot \hat{\nabla} \hat{h_{\mathfrak{m}}} - \sum_{\mathfrak{m}} \hat{h_{\mathfrak{m}}} \dot{\rho_{\mathfrak{m}}}. \end{aligned} \tag{A.25}$$

Here we defined $Pr = \mu_0 C_{p,\text{mix}_0} / \lambda_{T0}$, $\hat{\lambda_T} = \lambda_T / \lambda_{T0}$, and $C_{p,\text{mix}_0} = \bar{R}_{\text{gas}}$.

Again in the low Mach number limit, the leading order equation becomes

$$\hat{\rho}^0 \hat{C}^{\,0}_{p,\text{mix}} \frac{D\hat{T}^0}{D\hat{t}} = \frac{\partial \hat{p}^0}{\partial t} - \frac{1}{RePr} \hat{\nabla} \cdot \hat{\mathbf{q}}^0 + \sum_{\mathfrak{m}} \hat{\nabla} \cdot \left(\hat{\rho}^0 \hat{h_{\mathfrak{m}}}^{\,0} Y^0_{\mathfrak{m}} \hat{\mathbf{V}}_{\mathfrak{m}}^{\,0} \right) - \sum_{\mathfrak{m}} \hat{\rho}^0 Y^0_{\mathfrak{m}} \hat{\mathbf{V}}_{\mathfrak{m}}^{\,0} \cdot \hat{\nabla} \hat{h_{\mathfrak{m}}}^{\,0} - \sum_{\mathfrak{m}} \hat{h_{\mathfrak{m}}}^{\,0} \dot{\rho_{\mathfrak{m}}}. \tag{A.26}$$

The first term on the RHS drops for external flow where ambient pressure is constant in time.

A.2.4 Species transport equations

The species transport equation, Eqn. A.1d, does not change its form at the low Mach number limit. However, in the low Mach number limit, the pressure diffusion term in Eqn.(A.5) is dropped from the leading-order balance for the diffusion velocity. Further, if the thermal diffusion (Soret effect) is negligible, we can use a Fickian diffusion model:

$$Y_{\mathfrak{m}} \mathbf{V}_{\mathfrak{m}} = -D_{\mathfrak{m}} \nabla Y_{\mathfrak{m}}, \tag{A.27}$$

where $D_{\mathfrak{m}}$ is the mixture-averaged diffusion coefficient of species $\mathfrak{m}$.

It should be pointed out that the Soret effect can be of comparable order to the Fickian diffusion term (Ern & Giovangigli, 1998), if included, near the flame front where the

temperature gradient is large (Dimotakis, 2005). Therefore, computed flame thickness may contain some error. However, its influence on flame speed is not significant as shown by Ern & Giovangigli (1999), and the difference was mostly observed in rich flames in their study.

It is known that the inclusion of the Soret effect needs to be accomodated with a change in a wall boundary condition. $\mathbf{V}_\mathrm{m} = 0$ at the boundary does not necessarily translate to $\mathbf{n} \cdot \nabla Y_\mathrm{m} = 0$ on the boundary when additional diffusion terms are included. In particular, when iso-thermal wall boundary condition is used, the boundary conditions to species transport equations must be derived from Eqns. (A.4) and (A.5) to be consistent with a diffusion model.

A.2.5 Transport properties

For viscosity and other transport properties, we know μ and λ, or alternatively $\hat{\mu}$ and $\hat{\lambda}$ as a function of temperature θ. To solve the leading-order momentum equations, we need μ^0 and λ^0 as a function of θ^0. The Taylor series expansion gives us $\hat{\mu}(\hat{\theta}) = \mu^0(\theta^0) + O(\epsilon)$ and $\hat{\lambda}(\hat{\theta}) = \lambda^0(\theta^0) + O(\epsilon)$.

Even when the compositions (Y_ms) change, the same argument follows.

A.3 The governing equations used in this study

The low Mach number equations derived above can be written in the following form for equations in dimensional form:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{A.28a}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p^* + \frac{1}{\rho} \nabla \cdot \left(\mu \left[\nabla \mathbf{u} + \nabla \mathbf{u}^t \right] \right) + \mathbf{f}, \tag{A.28b}$$

$$\frac{\partial Y_\mathrm{m}}{\partial t} + \mathbf{u} \cdot \nabla Y_\mathrm{m} = -\frac{1}{\rho} \nabla \cdot (\rho Y_\mathrm{m} \mathbf{V}_\mathrm{m}) + \dot{\omega}_\mathrm{m}, \tag{A.28c}$$

$$\rho C_{p,\mathrm{mix}} \frac{DT}{Dt} = \nabla \cdot (\lambda_T \nabla T) - \rho \sum_\mathrm{m} h_\mathrm{m} \dot{\omega}_\mathrm{m} - \sum_\mathrm{m} \rho Y_\mathrm{m} \mathbf{V}_\mathrm{m} \cdot \nabla h_\mathrm{m}, \tag{A.28d}$$

$$\rho = p_0 / (\bar{R}_\mathrm{gas} T). \tag{A.28e}$$

Expand the viscous term of Eqn.(A.28b) to obtain

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p^* + \mathbf{L}^e(\mathbf{u}) + \mathbf{L}^i(\mathbf{u}) + \mathbf{f}, \tag{A.29}$$

where

$$\mathbf{L}^i(\mathbf{u}) = \frac{\mu}{\rho} \nabla^2 \mathbf{u} \tag{A.30a}$$

$$\mathbf{L}^e(\mathbf{u}) = \frac{\mu}{\rho} \nabla \left[\nabla \cdot \mathbf{u} \right] + \frac{\nabla \mu}{\rho} \cdot \left[\nabla \mathbf{u} + \nabla \mathbf{u}^t \right] \tag{A.30b}$$

This particular form of the momentum equation is integrated. One could combine the axial and radial momentum equations to obtain the fully coupled weak formulation of the momentum equations that allows the entire viscous term to be treated implicitly. However, for efficiency reasons, the above approach is taken in this study. The viscous term is further split into the explicit part and the implicit part to save computational time in forming the stiffness matrix used in the implicit integration of the viscous term. Details can be found in Chapter 3.

For the species transport equations, $\mathbf{V}_{\mathfrak{m}}$ needs to be modeled, and the Fickian diffusion model is used in this study.

The diffusion velocity of species $\mathfrak{m}$ must satisfy the following:

$$\sum_{\mathfrak{m}} Y_{\mathfrak{m}} \mathbf{V}_{\mathfrak{m}} = 0. \tag{A.31}$$

This constraint is the result of mass conservation. However, a drawback of using the Fickian diffusion model is that this constraint on $\mathbf{V}_{\mathfrak{m}}$ may not be satisifed. A technique proposed by Coffee & Heimerl (1981) defines a corrective differential velocity, V_c such that

$$\mathbf{V}_c = -\sum_{\mathfrak{m}} Y_{\mathfrak{m}} \tilde{\mathbf{V}}_{\mathfrak{m}}, \tag{A.32}$$

where $\tilde{\mathbf{V}}_{\mathfrak{m}}$ now follows the Fickian model:

$$\tilde{\mathbf{V}}_{\mathfrak{m}} = -\frac{D_{\mathfrak{m}}}{Y_{\mathfrak{m}}} \nabla Y_{\mathfrak{m}}, \tag{A.33}$$

so that

$$\mathbf{V}_{\mathfrak{m}} = \tilde{\mathbf{V}}_{\mathfrak{m}} + \mathbf{V}_c, \tag{A.34}$$

therefore

$$\sum_{\mathfrak{m}} Y_{\mathfrak{m}} \mathbf{V}_{\mathfrak{m}} = 0. \tag{A.35}$$

Effectively, this method subtracts the correct amount of excessive diffusion distributed equally among all species existing in the system. For example, Day & Bell (2000) used this algorithm in their study. However, use of this algorithm in an implicit method is not practical because Eqn.(A.32) couples the entire species through summation. If one of the species is inert and abundant,

$$V_{\mathfrak{m}'} = \frac{1}{Y_{\mathfrak{m}'}} \left(\sum_{\mathfrak{m} \in \mathcal{M}'} D_{\mathfrak{m}} \nabla Y_{\mathfrak{m}} \right), \tag{A.36}$$

where $\mathfrak{m}'$ is the abundant species (nitrogen in this study), and the set $\mathcal{M}'$ contains all the other species but $\mathfrak{m}'$. This is a more practical method to satisfy the constraint without sacrificing efficiency.

Again, similar to the momentum equation, the diffusion term is expanded into an explicit part and an implicit part:

$$\frac{\partial Y_{\mathfrak{m}}}{\partial t} + \mathbf{u} \cdot \nabla Y_{\mathfrak{m}} = D_{\mathfrak{m}} \nabla^2 Y_{\mathfrak{m}} + \frac{1}{\rho} \nabla \rho D_{\mathfrak{m}} \cdot \nabla Y_{\mathfrak{m}} + \dot{\omega}_{\mathfrak{m}} \tag{A.37}$$

for $\mathfrak{m} \in \mathcal{M}'$. It should be noted that as a result of the constraint Eqn.(A.36) and the mass conservation, Y_{N_2} is uniquely determined by the constraint, Eqn.(2.50).

When the Fickian diffusion model is employed, the last term in Eqn.(A.28d) disappears. The radiant heat flux is ignored because it has a noticable effect only at the lean flammability limits (Egolfopoulos, 1994). The Dufour heat flux is also negligible (Ern & Giovangigli, 1998; Williams, 1985). By putting those assumptions into the low Mach number limit of the temperature transport equations, we obtain,

$$\rho C_{p,\mathrm{mix}} \frac{DT}{Dt} = \lambda_T \nabla^2 T + \nabla \lambda_T \cdot \nabla T - \rho \sum_{\mathfrak{m}} h_{\mathfrak{m}} \dot{\omega}_{\mathfrak{m}}. \tag{A.38}$$

Again, the diffusion term has been split into explicit and implicit parts.

A.4 The differential form of the equations in cylindrical coordinates

Vector notation is used throughout this thesis, however, an explicit derivative formulation is necessary to implement the evaluation of derivatives. The differential form of various tensor expressions in cylindrical coordinates can be found in Lagerstrom (1964) and Emmons (1958).

Let (r, θ, z) be the radial, azimuthal, and axial coordinates of the cylindrical polar coordinate respectively, and (u_r, u_θ, u_z) be the corresponding velocity vector. To compute the stress tensor in cylindrical coordinates, we need the following:

$$\nabla \mathbf{u} = \begin{pmatrix} \frac{\partial u_r}{\partial r} & \frac{1}{r}\frac{\partial u_r}{\partial \theta} - \frac{u_\theta}{r} & \frac{\partial u_r}{\partial z} \\ \frac{\partial u_\theta}{\partial r} & \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r} & \frac{\partial u_\theta}{\partial z} \\ \frac{\partial u_z}{\partial r} & \frac{1}{r}\frac{\partial u_z}{\partial \theta} & \frac{\partial u_z}{\partial z} \end{pmatrix} \tag{A.39}$$

and let $\mathbf{T}$ be a 3-dimensional tensor of rank 2, then

$$\nabla \cdot \mathbf{T} = \begin{pmatrix} \frac{\partial \tau_{rr}}{\partial r} + \frac{\tau_{rr}}{r} + \frac{1}{r}\frac{\partial \tau_{r\theta}}{\partial \theta} - \frac{\tau_{\theta\theta}}{r} + \frac{\partial \tau_{rz}}{\partial z} \\ \frac{\partial \tau_{\theta r}}{\partial r} + \frac{\tau_{\theta r}}{r} + \frac{1}{r}\frac{\partial \tau_{\theta\theta}}{\partial \theta} + \frac{\tau_{r\theta}}{r} + \frac{\partial \tau_{\theta z}}{\partial z} \\ \frac{\partial \tau_{zr}}{\partial r} + \frac{\tau_{zr}}{r} + \frac{1}{r}\frac{\partial \tau_{z\theta}}{\partial \theta} + \frac{\partial \tau_{zz}}{\partial z} \end{pmatrix}. \tag{A.40}$$

Therefore, the viscous stress tensor, $\mathbf{T} = \mu[(\nabla \mathbf{u}) + (\nabla \mathbf{u})^T]$ has the following form,

$$\mathbf{T} = \begin{pmatrix} 2\mu\frac{\partial u_r}{\partial r} & \mu\left(\frac{1}{r}\frac{\partial u_r}{\partial \theta} - \frac{u_\theta}{r} + \frac{\partial u_\theta}{\partial r}\right) & \mu\left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r}\right) \\ \mu\left(\frac{1}{r}\frac{\partial u_r}{\partial \theta} - \frac{u_\theta}{r} + \frac{\partial u_\theta}{\partial r}\right) & 2\mu\left(\frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r}\right) & \mu\left(\frac{\partial u_\theta}{\partial z} + \frac{1}{r}\frac{\partial u_z}{\partial \theta}\right) \\ \mu\left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r}\right) & \mu\left(\frac{\partial u_\theta}{\partial z} + \frac{1}{r}\frac{\partial u_z}{\partial \theta}\right) & 2\mu\frac{\partial u_z}{\partial z} \end{pmatrix}. \tag{A.41}$$

So that, the viscous term in the momentum equation is,

$$\nabla\cdot\mathbf{T}=\left(\begin{array}{l}\frac{1}{r}\frac{\partial}{\partial r}\left(2\mu r\frac{\partial u_r}{\partial r}\right)+\frac{1}{r}\frac{\partial}{\partial\theta}\left(\mu\left(\frac{1}{r}\frac{\partial u_r}{\partial\theta}-\frac{u_\theta}{r}+\frac{\partial u_\theta}{\partial r}\right)\right)-\frac{1}{r}\left(2\mu\left(\frac{1}{r}\frac{\partial u_\theta}{\partial\theta}+\frac{u_r}{r}\right)\right)\\ \qquad\qquad+\frac{\partial}{\partial z}\left(\mu\left(\frac{\partial u_r}{\partial z}+\frac{\partial u_z}{\partial r}\right)\right)\\ \frac{1}{r}\frac{\partial}{\partial r}\left(\mu r\left(\frac{1}{r}\frac{\partial u_r}{\partial\theta}-\frac{u_\theta}{r}+\frac{\partial u_\theta}{\partial r}\right)\right)+\frac{1}{r}\frac{\partial}{\partial\theta}\left(2\mu\left(\frac{1}{r}\frac{\partial u_\theta}{\partial\theta}+\frac{u_r}{r}\right)\right)\\ \qquad\qquad+\frac{1}{r}\left(\mu\left(\frac{1}{r}\frac{\partial u_r}{\partial\theta}-\frac{u_\theta}{r}+\frac{\partial u_\theta}{\partial r}\right)\right)+\frac{\partial}{\partial z}\left(\mu\left(\frac{\partial u_\theta}{\partial z}+\frac{1}{r}\frac{\partial u_z}{\partial\theta}\right)\right)\\ \frac{1}{r}\frac{\partial}{\partial r}\left(\mu r\left(\frac{\partial u_r}{\partial z}+\frac{\partial u_z}{\partial r}\right)\right)+\frac{1}{r}\frac{\partial}{\partial\theta}\left(\mu\left(\frac{\partial u_\theta}{\partial z}+\frac{1}{r}\frac{\partial u_z}{\partial\theta}\right)\right)+\frac{\partial}{\partial z}\left(2\mu\frac{\partial u_z}{\partial z}\right)\end{array}\right),\tag{A.42}$$

which is in the case of axisymmetry,

$$\nabla\cdot\mathbf{T}=\left(\begin{array}{l}\frac{1}{r}\frac{\partial}{\partial r}\left(2\mu r\frac{\partial u_r}{\partial r}\right)-2\mu\frac{u_r}{r^2}+\frac{\partial}{\partial z}\left(\mu\left(\frac{\partial u_r}{\partial z}+\frac{\partial u_z}{\partial r}\right)\right)\\ 0\\ \frac{1}{r}\frac{\partial}{\partial r}\left(\mu r\left(\frac{\partial u_r}{\partial z}+\frac{\partial u_z}{\partial r}\right)\right)+\frac{\partial}{\partial z}\left(2\mu\frac{\partial u_z}{\partial z}\right)\end{array}\right).\tag{A.43}$$

Similarly, the nonlinear terms in the momentum equations are

$$\mathbf{u}\cdot\nabla\mathbf{u}=\left(\begin{array}{c}u_r\frac{\partial u_r}{\partial r}+\frac{u_\theta}{r}\frac{\partial u_r}{\partial\theta}-\frac{u_\theta^2}{r}+u_z\frac{\partial u_r}{\partial z}\\ u_r\frac{\partial u_\theta}{\partial r}+\frac{u_\theta}{r}\frac{\partial u_\theta}{\partial\theta}+\frac{u_r u_\theta}{r}+u_z\frac{\partial u_\theta}{\partial z}\\ u_r\frac{\partial u_z}{\partial r}+\frac{u_\theta}{r}\frac{\partial u_z}{\partial\theta}+u_z\frac{\partial u_z}{\partial z}\end{array}\right),\tag{A.44}$$

which is in the axisymmetric case,

$$\mathbf{u}\cdot\nabla\mathbf{u}=\left(\begin{array}{c}u_r\frac{\partial u_r}{\partial r}+u_z\frac{\partial u_r}{\partial z}\\ 0\\ u_r\frac{\partial u_z}{\partial r}+u_z\frac{\partial u_z}{\partial z}\end{array}\right).\tag{A.45}$$

The vector equation (A.28b) is comprised of two equations in differential form,

$$\begin{aligned}\frac{\partial u_z}{\partial t}+\left(u_z\frac{\partial u_z}{\partial z}+u_r\frac{\partial u_z}{\partial r}\right)=&-\frac{1}{\rho}\frac{\partial p^*}{\partial z}\\ &+\frac{1}{\rho}\left\{\frac{1}{r}\frac{\partial}{\partial r}\left(\mu r\left(\frac{\partial u_r}{\partial z}+\frac{\partial u_z}{\partial r}\right)\right)+\frac{\partial}{\partial z}\left(2\mu\frac{\partial u_z}{\partial z}\right)\right\}+f_z\end{aligned}\tag{A.46}$$

and

$$\frac{\partial u_r}{\partial t} + \left(u_z \frac{\partial u_r}{\partial z} + u_r \frac{\partial u_r}{\partial r} \right) = -\frac{1}{\rho} \frac{\partial p^*}{\partial r} + \frac{1}{\rho} \left\{ \frac{1}{r} \frac{\partial}{\partial r} \left(2\mu r \frac{\partial u_r}{\partial r} \right) - 2\mu \frac{u_r}{r^2} + \frac{\partial}{\partial z} \left(\mu \left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \right) \right) \right\} + f_r \tag{A.47}$$

for the cylindrical coordinates when the flow is axisymmetric ($\partial/\partial\theta = 0$) and there is no swirl ($u_\theta = 0$), both of which are assumed in this study.

Appendix B

Initial conditions to unsteady flame simulations

B.1 Introduction

Although the initial conditions to a time-dependent problem are important, not much attention has been paid to this aspect. Unfortunately, many multidimensional numerical studies on reacting flows that employed unsteady equations—Day & Bell (2000); Tomboulides *et al.* (1997); Frouzakis *et al.* (1998)—did not clearly describe the initial condition in their paper. In the present study, a one-dimensional solution using Cantera is used to create an initial condition for the Phase II flame simulation as follows:

$$u(z,r) = f(r;0.00425,0.0025)u^{\text{Cantera}}(z) \tag{B.1a}$$

$$v(z,r) = f(r;0.00425,0.0025)rv^{\text{Cantera}}(z) \tag{B.1b}$$

$$T(z,r) = T^{\text{Cantera}}(z) \tag{B.1c}$$

$$Y_{\mathfrak{m}}(z,r) = f(r;0.005,0.0005)Y_{\mathfrak{m}} \tag{B.1d}$$

$$\tag{B.1e}$$

where $f(r;r_0,\delta r)$ is a blending function that takes a value between 0 and 1, and is defined by

$$f(r;r_0,\delta r) = \left\{\tanh\left[\frac{2.64665}{\delta r/2}(r_0-r)\right]+1\right\}/2 \tag{B.2}$$

where the value of 2.64665 is obtained by $\tanh^{-1}(0.99)$. In other words, r_0 specifies the half-value location, and δr specifies the width of the 99 % window.

In obtaining the one-dimensional solution using Cantera, error torelances and grid refine-

ment parameters are tightened to obtain the numerical solution on a successively increased number of points, starting from around 100. The final solution is solved on more than 1,000 points.

Then, the initial conditions to the Phase III simulations are obtained by combining two different numerical solutions. One is the Phase II solution. The other is the cold flow numerical solution of the nozzle interior flow with the nozzle filled by the mixture at the inflow condition.

B.2 Conversion tables

Throughout this study, air is assumed to be composed of 21% Oxygen and 79% Nitrogen by volume, which is $Y_{\text{oxygen}} = 0.232918$ and $Y_{\text{nitrogen}} = 0.767082$. Table B.1 tabulates mole fractions and mass fractions of Methane, Oxygen, and Nitrogen at various equivalence ratios. Note the equivalence ratios tabulated here are accurate up to the sixth digit, *i.e.*, 1.30 should read 1.300000.

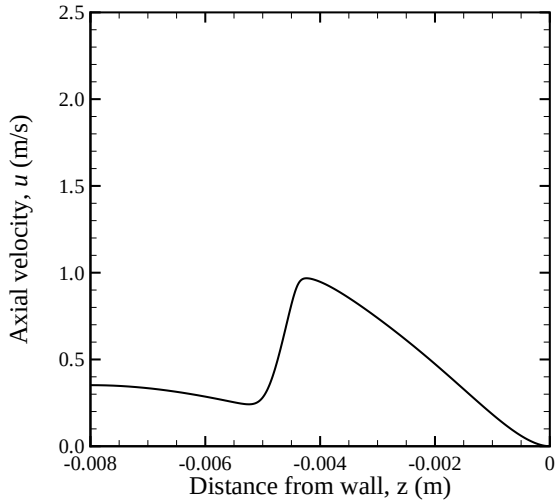
Table B.1: Methane / Air flame equivalence ratio to mass fraction

Φ	X_{CH_4}	X_{O_2}	X_{N_2}	Y_{CH_4}	Y_{O_2}	Y_{N_2}
0.50	0.049881	0.199525	0.750594	0.028366	0.226311	0.745324
0.55	0.054597	0.198535	0.746868	0.031114	0.225671	0.743215
0.60	0.059266	0.197554	0.743180	0.033847	0.225034	0.741119
0.65	0.063890	0.196583	0.739527	0.036564	0.224401	0.739035
0.70	0.068468	0.195622	0.735911	0.039266	0.223772	0.736962
0.75	0.073001	0.194670	0.732329	0.041954	0.223146	0.734901
0.80	0.077491	0.193727	0.728782	0.044626	0.222523	0.732851
0.85	0.081937	0.192793	0.725270	0.047283	0.221905	0.730813
0.90	0.086341	0.191868	0.721791	0.049925	0.221289	0.728786
0.95	0.090702	0.190952	0.718345	0.052553	0.220677	0.726770
1.00	0.095023	0.190045	0.714932	0.055167	0.220068	0.724765
1.05	0.099302	0.189147	0.711551	0.057766	0.219463	0.722772
1.10	0.103541	0.188256	0.708203	0.060350	0.218861	0.720789
1.15	0.107740	0.187375	0.704885	0.062921	0.218262	0.718817
1.20	0.111901	0.186501	0.701599	0.065477	0.217667	0.716856
1.25	0.116022	0.185635	0.698343	0.068020	0.217074	0.714905
1.30	0.120106	0.184778	0.695117	0.070549	0.216485	0.712966
1.35	0.124152	0.183928	0.691920	0.073064	0.215900	0.711036
1.40	0.128160	0.183086	0.688753	0.075566	0.215317	0.709117
1.45	0.132133	0.182252	0.685615	0.078054	0.214737	0.707209
1.50	0.136069	0.181425	0.682505	0.080529	0.214161	0.705310

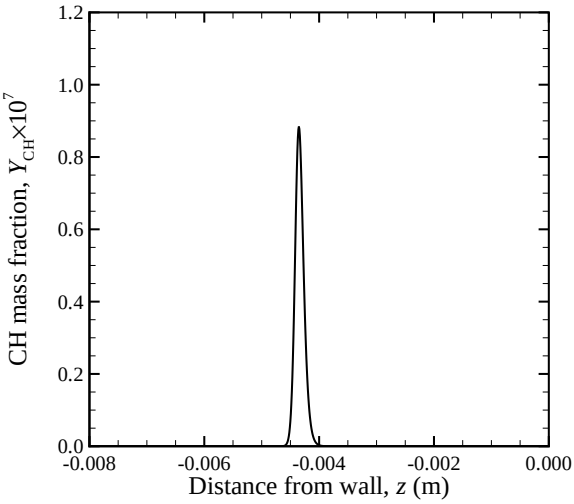
Appendix C

Flame simulation data

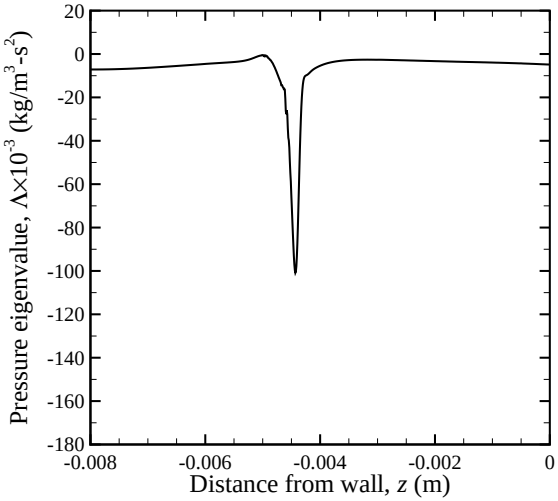
C.1 Phase II: 2D numerical experiments in cylindrical domain



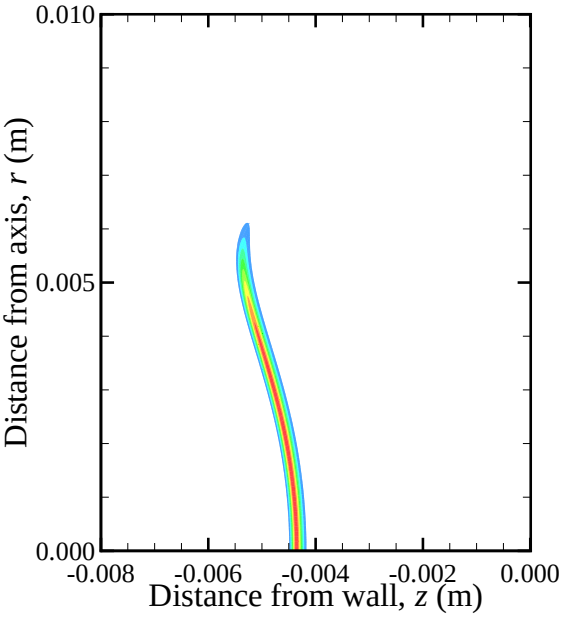
(a) Axial velocity profile



(b) CH mass fraction

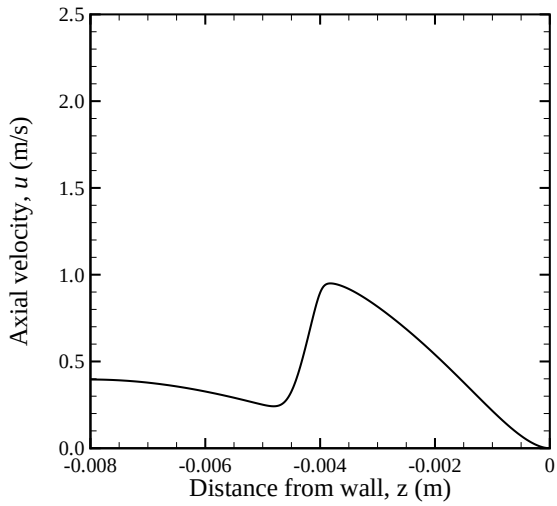


(c) Pressure eigenvalue

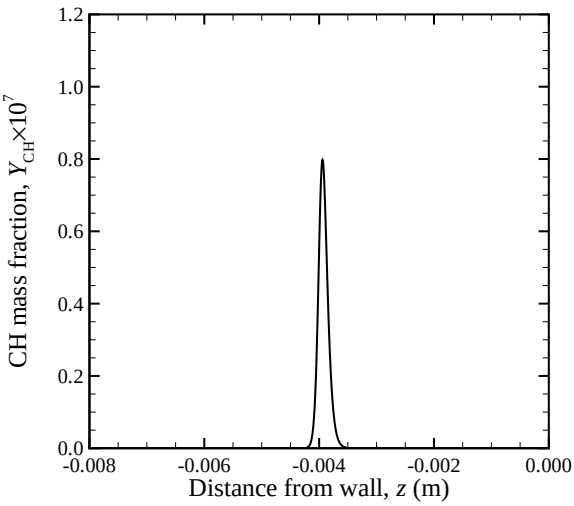


(d) CH contour

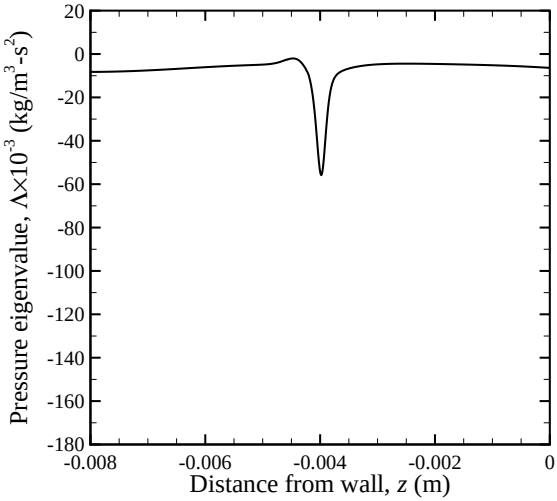
Figure C.1: Φ=0.70, Case B070-2



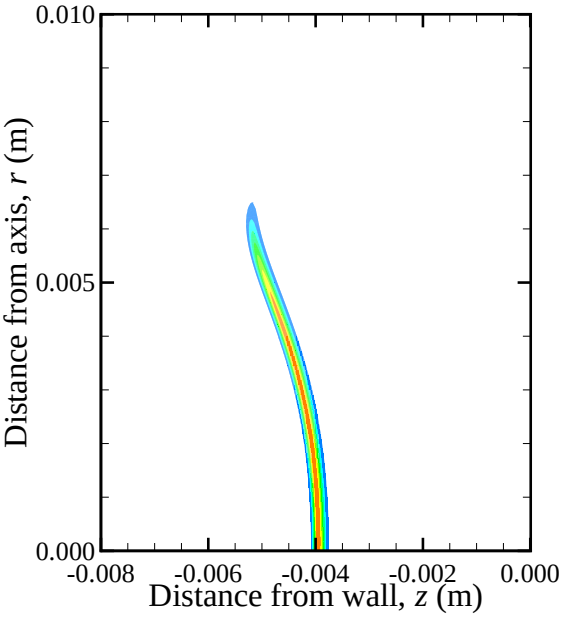
(a) Axial velocity profile



(b) CH mass fraction

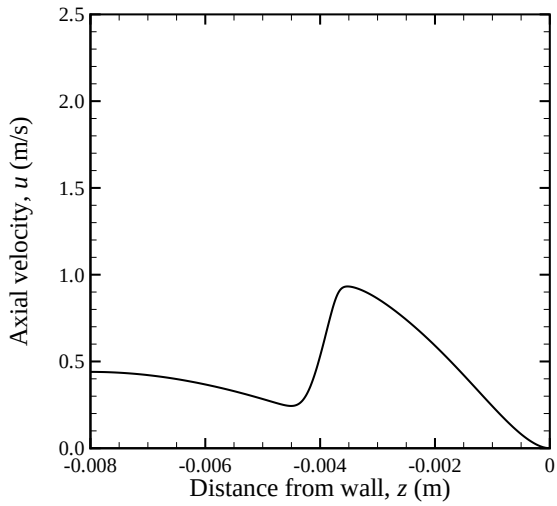


(c) Pressure eigenvalue

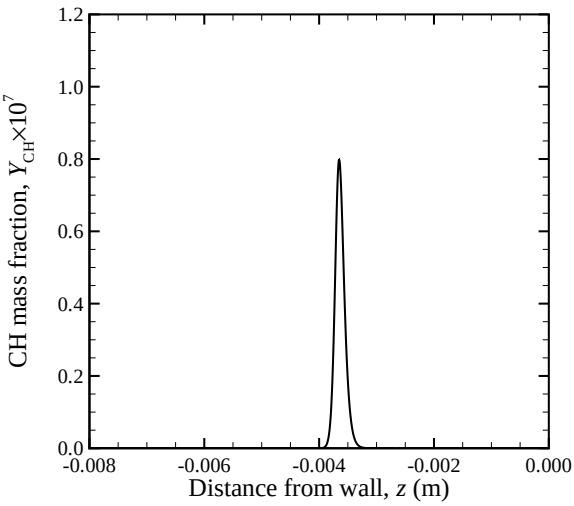


(d) CH contour

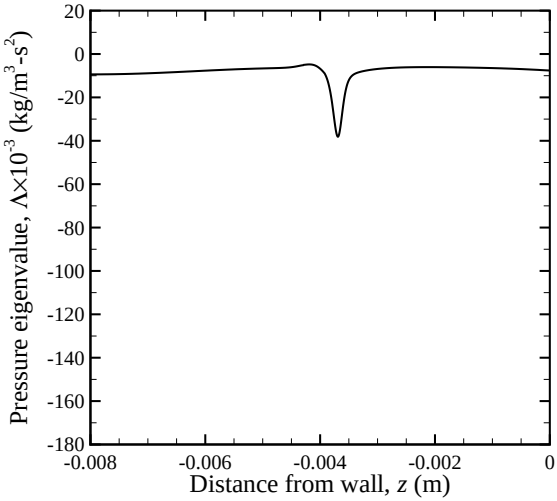
Figure C.2: Φ=0.70, Case B070-7



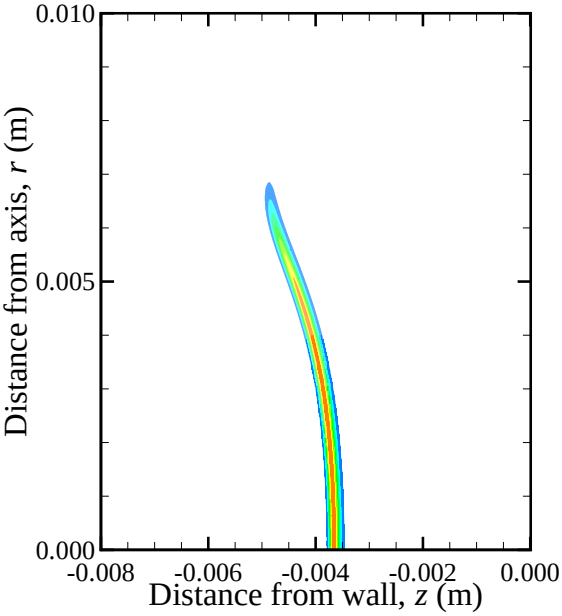
(a) Axial velocity profile



(b) CH mass fraction

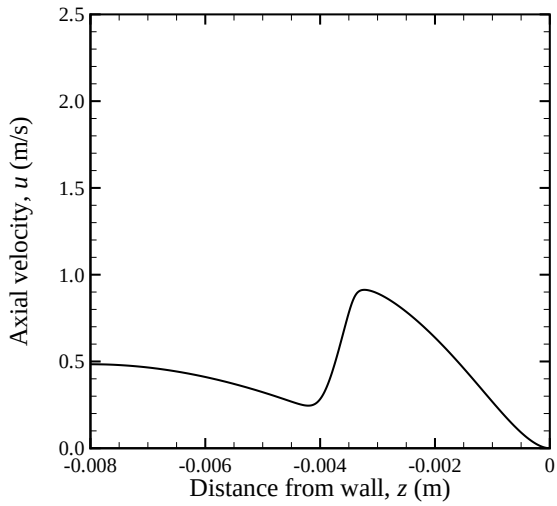


(c) Pressure eigenvalue

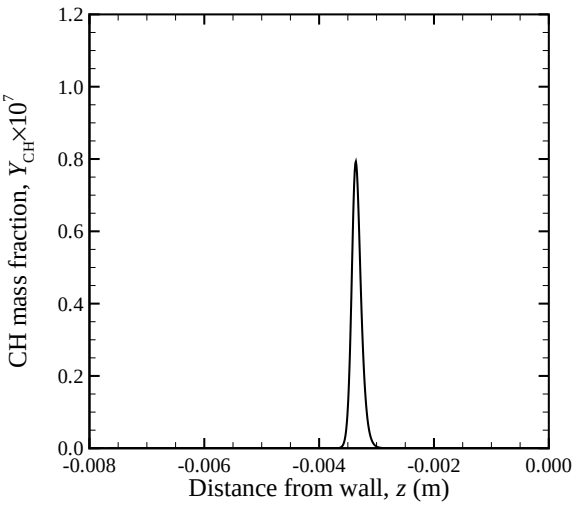


(d) CH contour

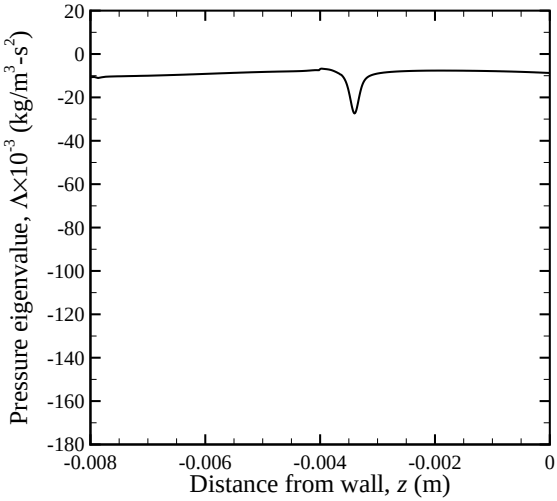
Figure C.3: Φ=0.70, Case B070-12



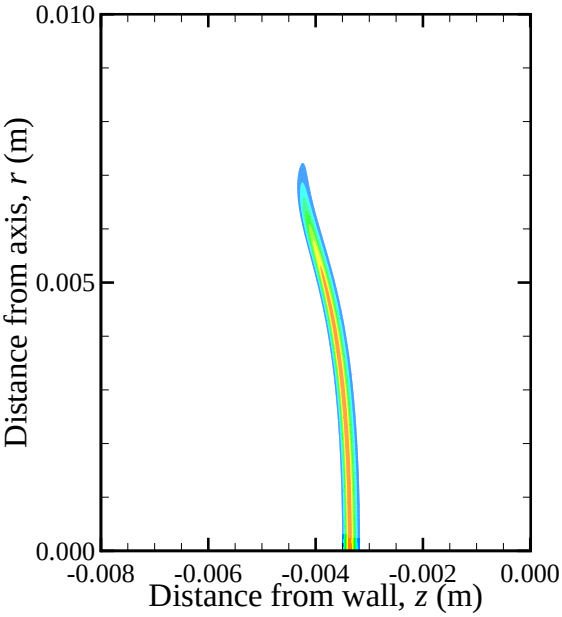
(a) Axial velocity profile



(b) CH mass fraction

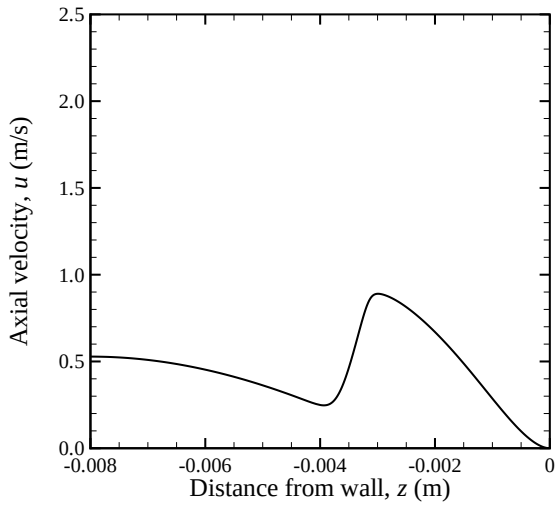


(c) Pressure eigenvalue

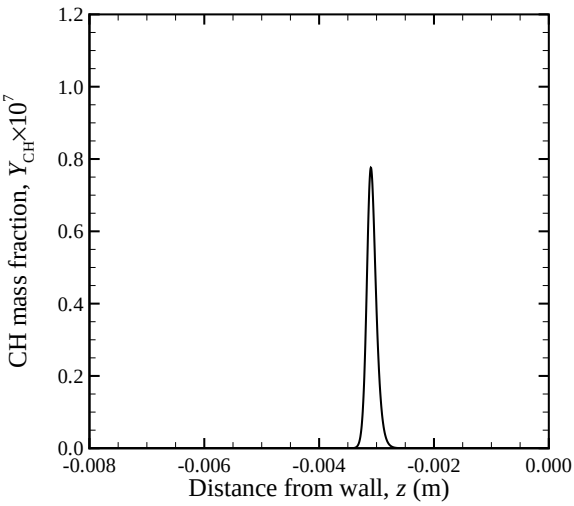


(d) CH contour

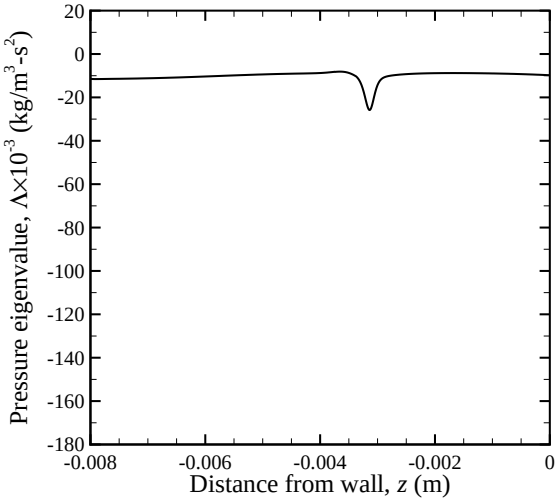
Figure C.4: Φ=0.70, Case B070-17



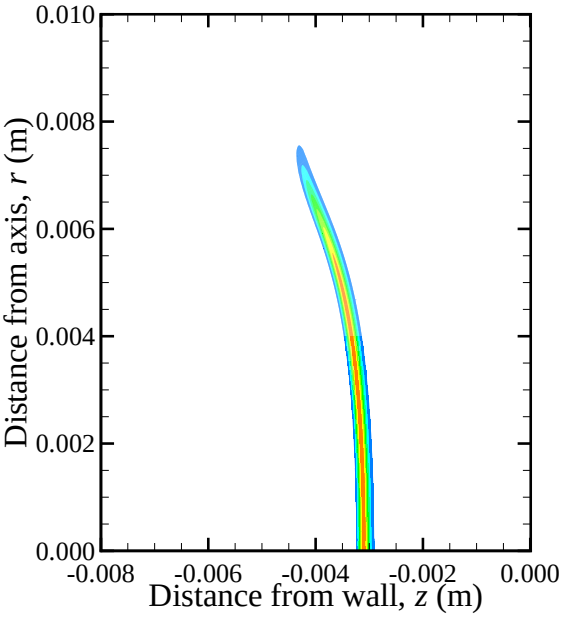
(a) Axial velocity profile



(b) CH mass fraction

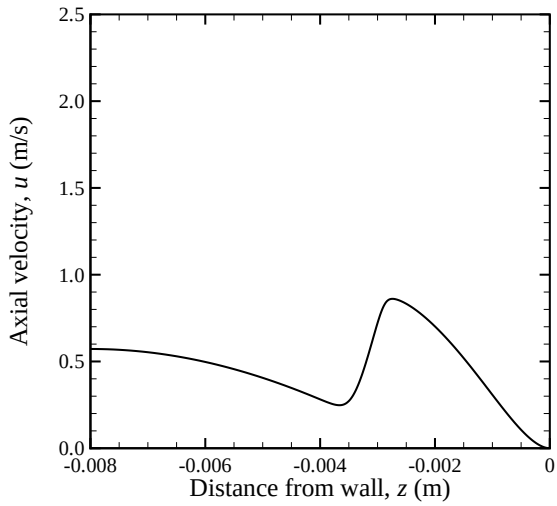


(c) Pressure eigenvalue

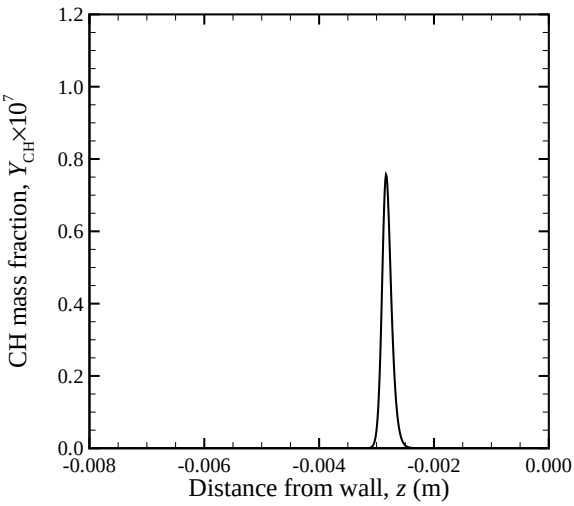


(d) CH contour

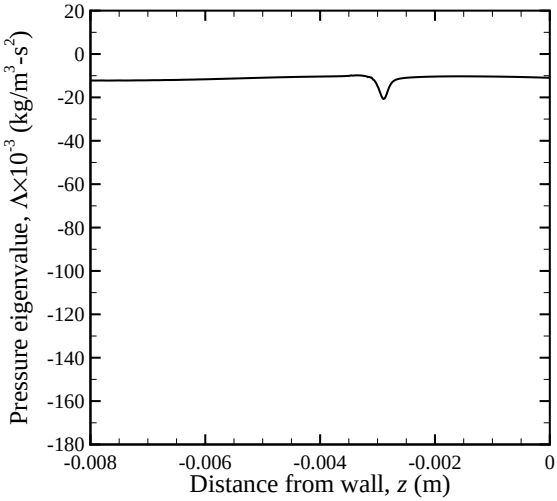
Figure C.5: Φ=0.70, Case B070-22



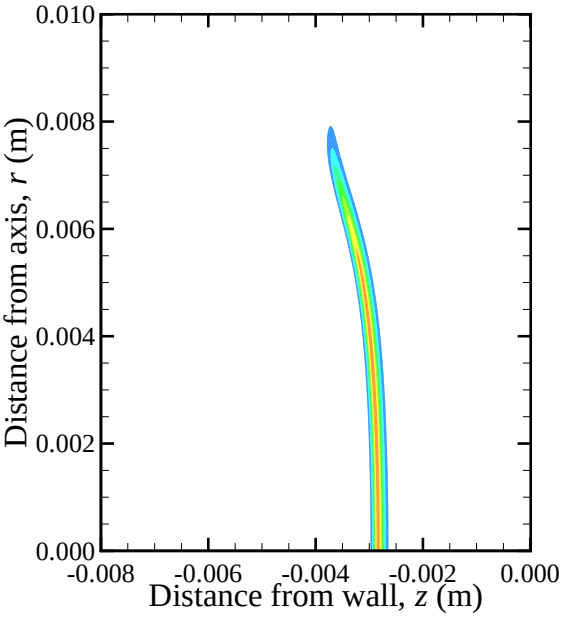
(a) Axial velocity profile



(b) CH mass fraction

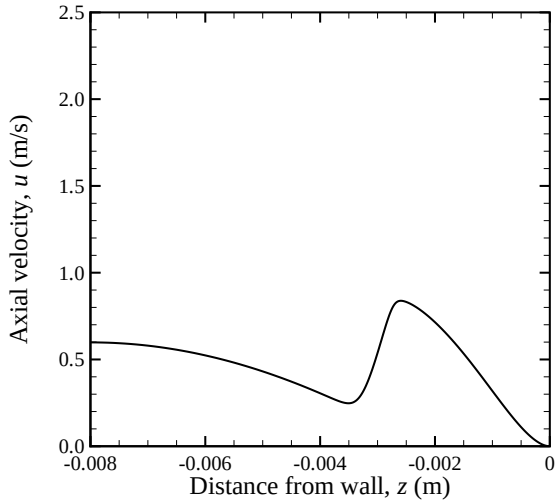


(c) Pressure eigenvalue

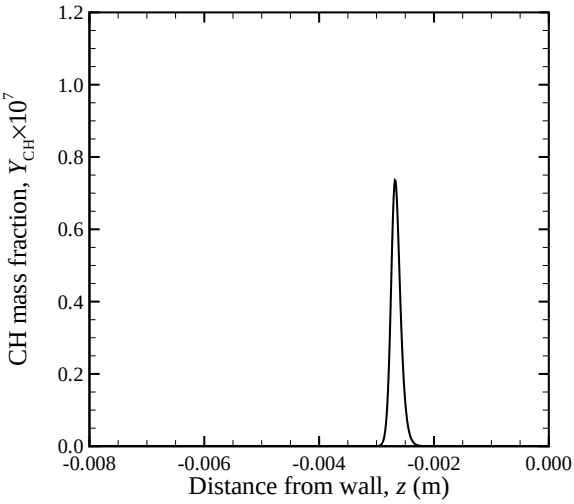


(d) CH contour

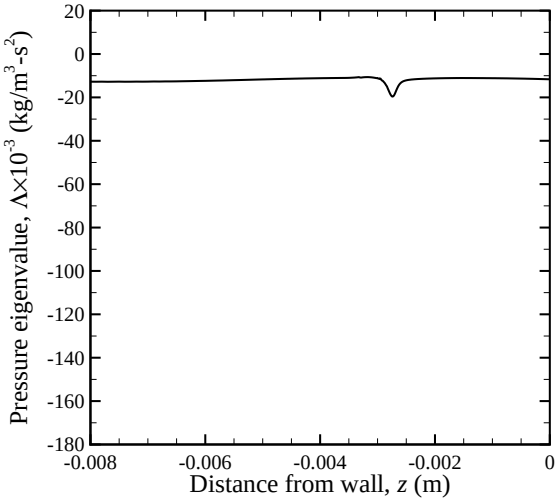
Figure C.6: Φ=0.70, Case B070-27



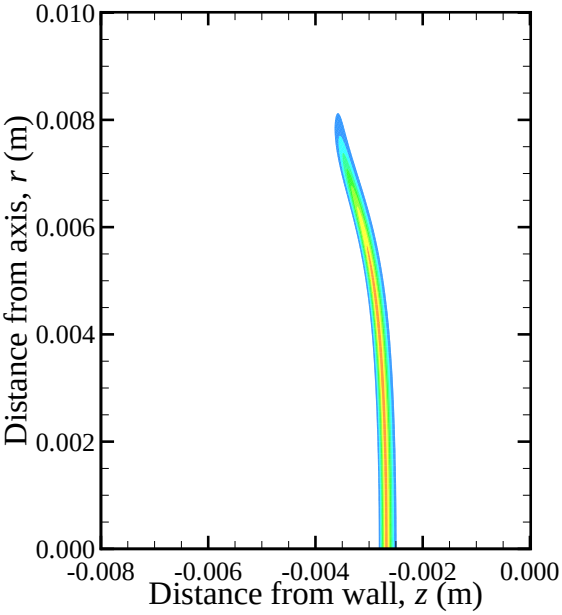
(a) Axial velocity profile



(b) CH mass fraction

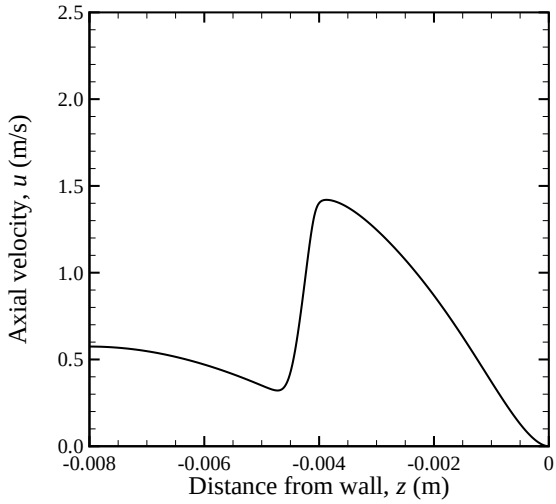


(c) Pressure eigenvalue

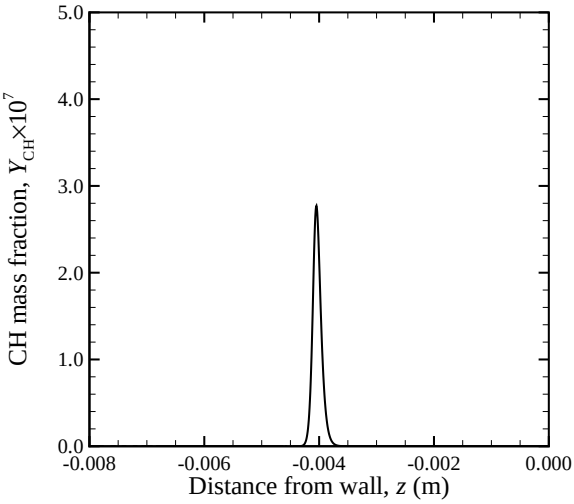


(d) CH contour

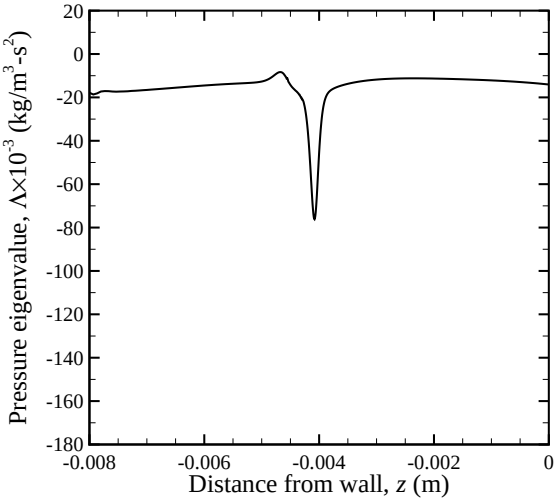
Figure C.7: Φ=0.70, Case B070-30



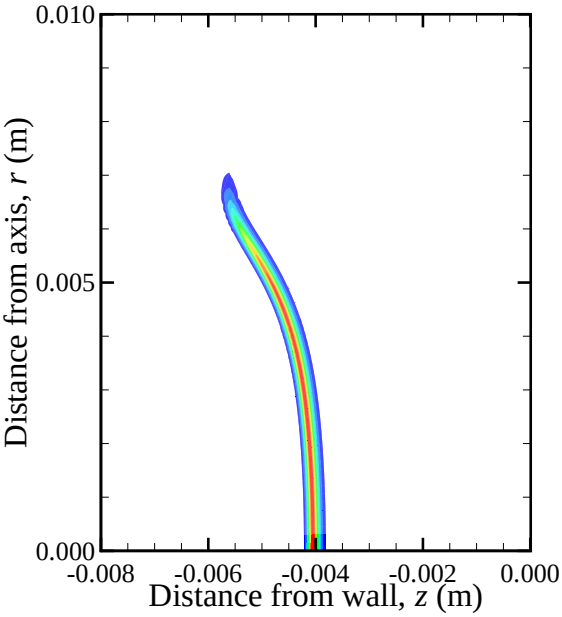
(a) Axial velocity profile



(b) CH mass fraction

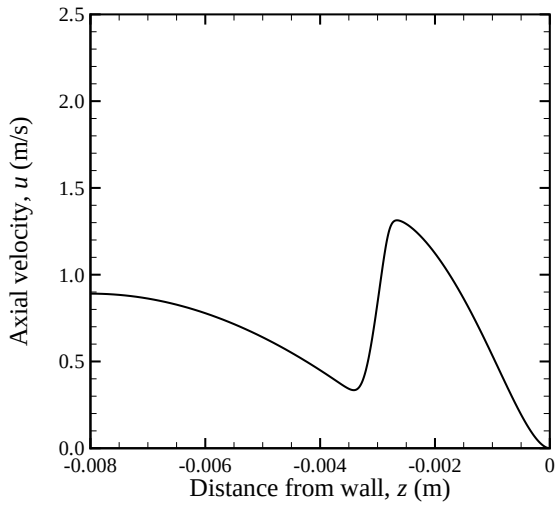


(c) Pressure eigenvalue

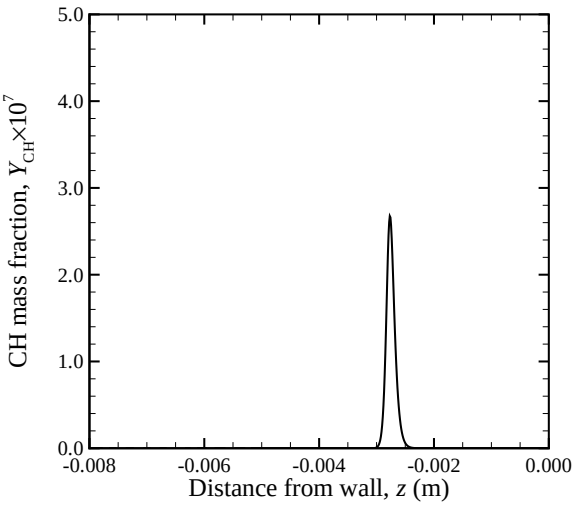


(d) CH contour

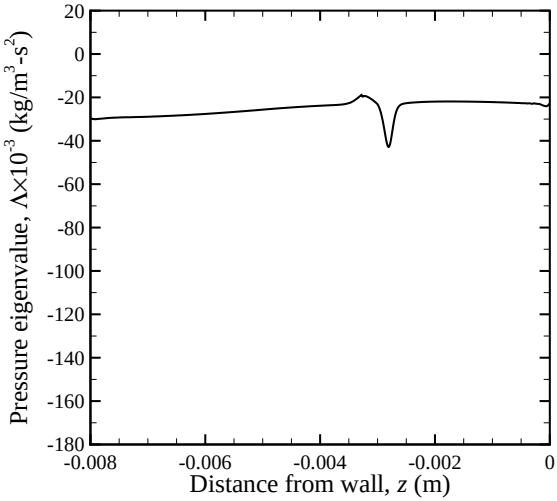
Figure C.8: Φ=0.80, Case B080-17



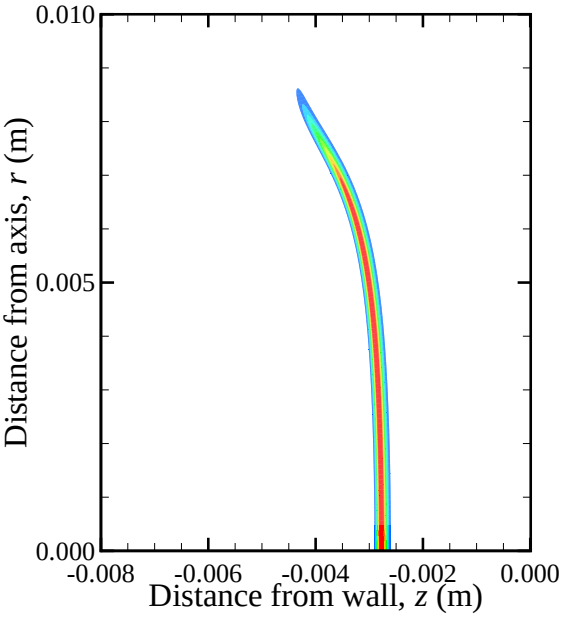
(a) Axial velocity profile



(b) CH mass fraction

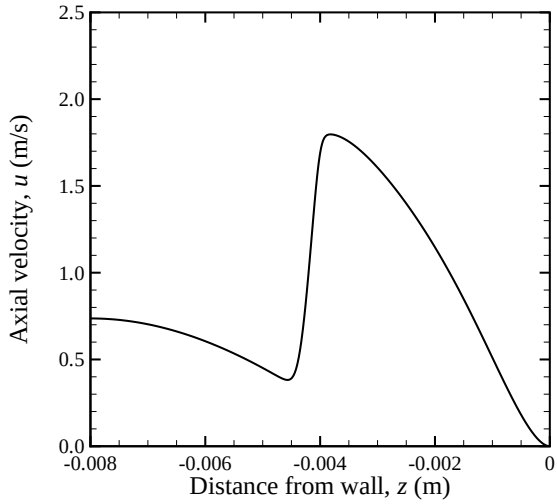


(c) Pressure eigenvalue

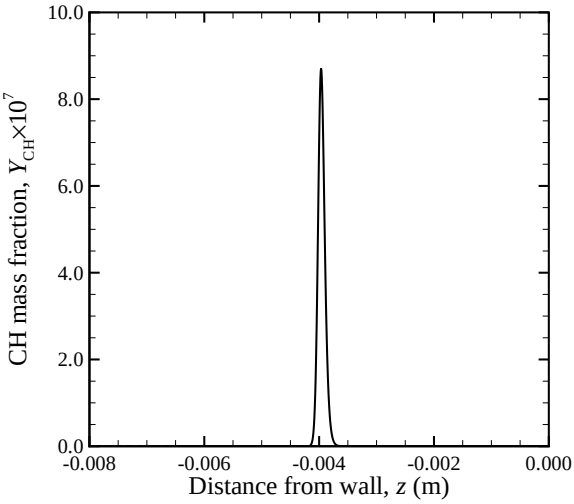


(d) CH contour

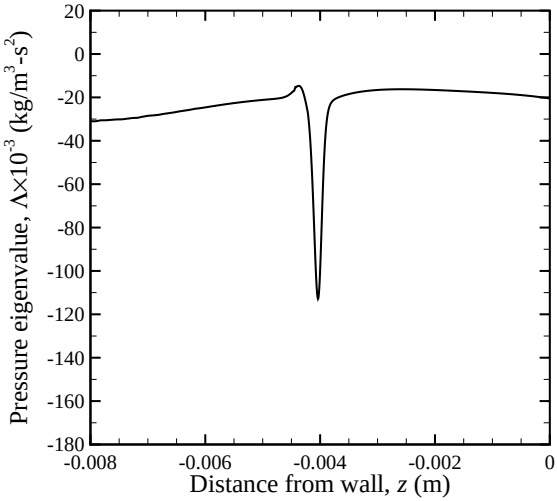
Figure C.9: Φ=0.80, Case B080-46



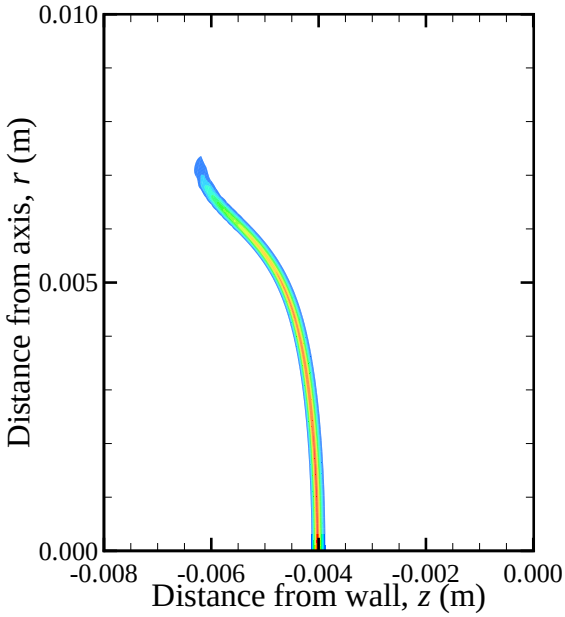
(a) Axial velocity profile



(b) CH mass fraction

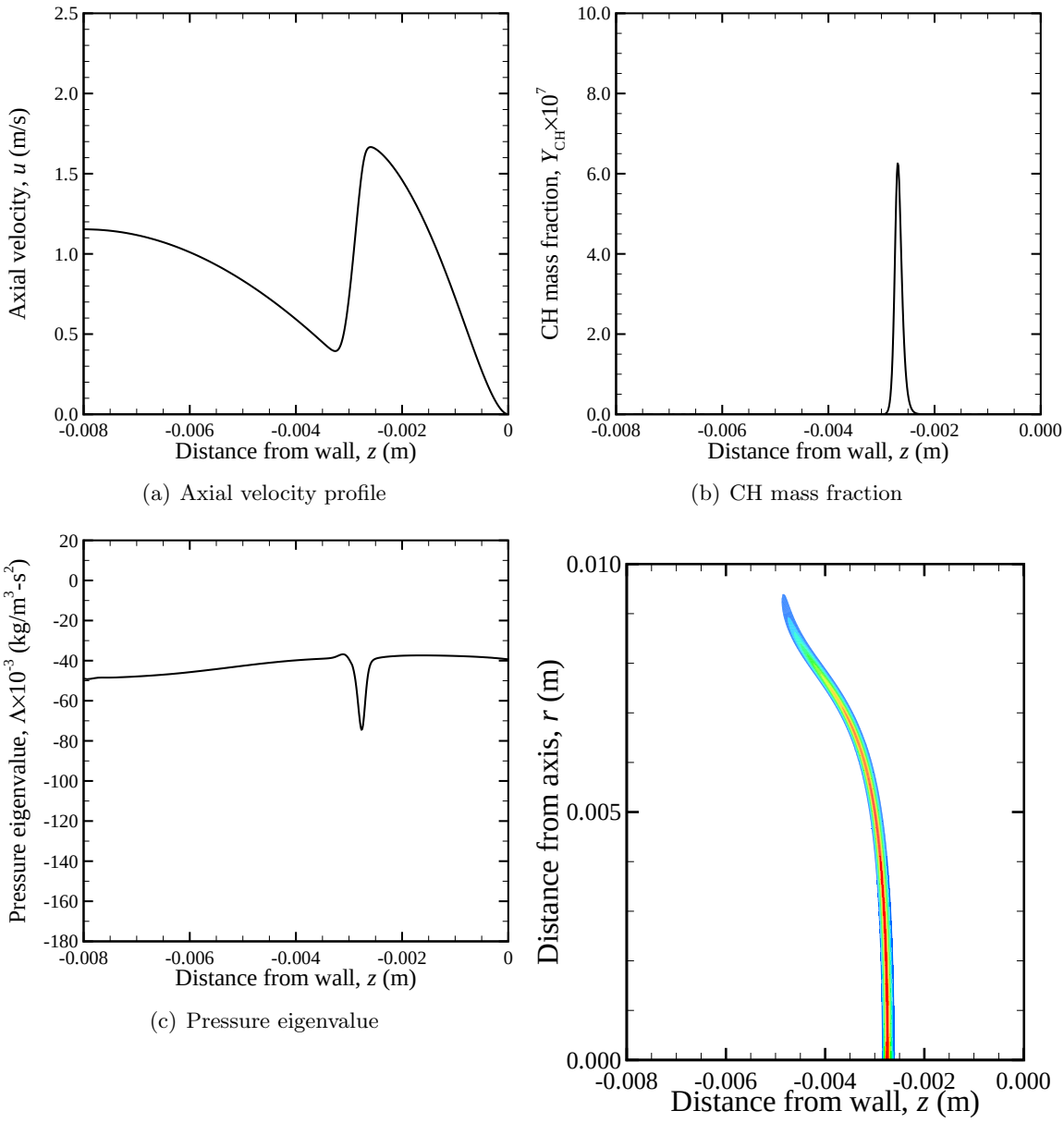


(c) Pressure eigenvalue



(d) CH contour

Figure C.10: Φ=0.90, Case B090-30



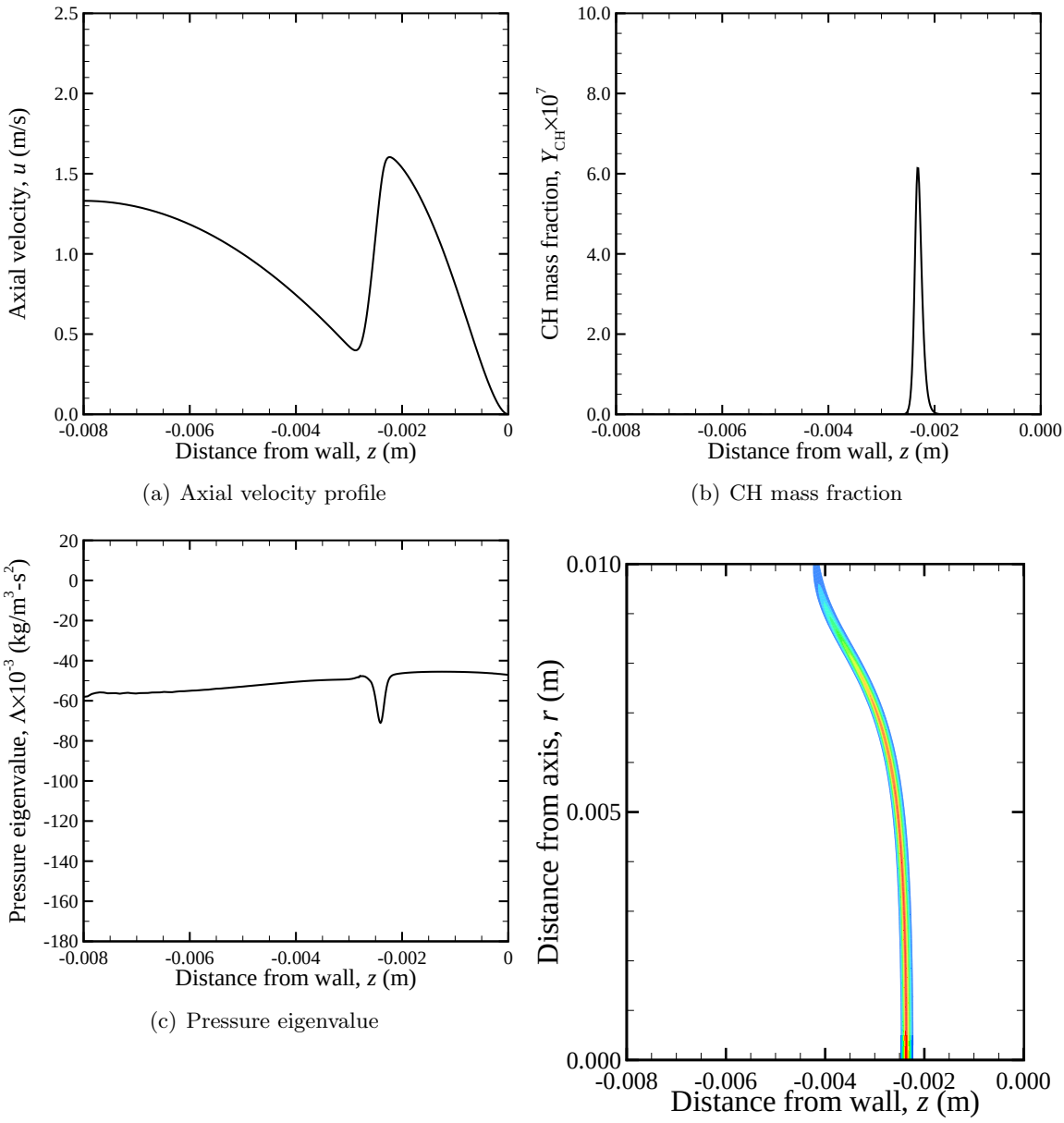
(a) Axial velocity profile

(b) CH mass fraction

(c) Pressure eigenvalue

(d) CH contour

Figure C.11: Φ=0.90, Case B090-50



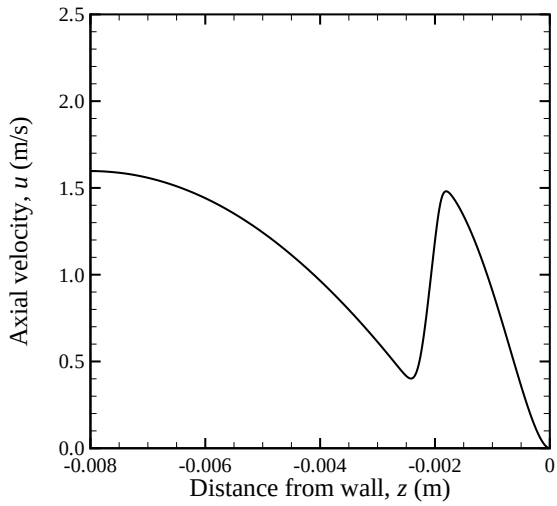
(a) Axial velocity profile

(b) CH mass fraction

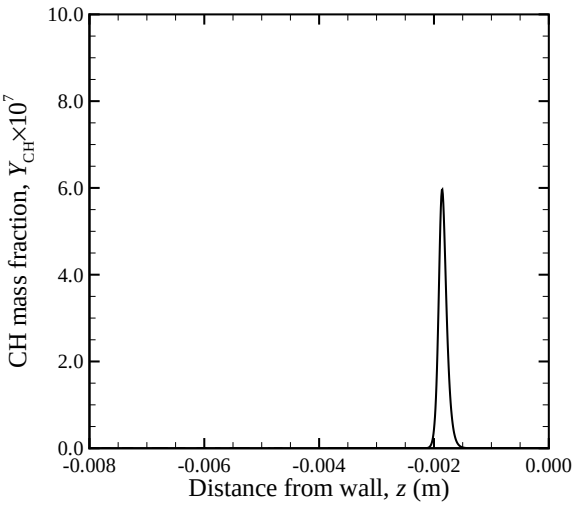
(c) Pressure eigenvalue

(d) CH contour

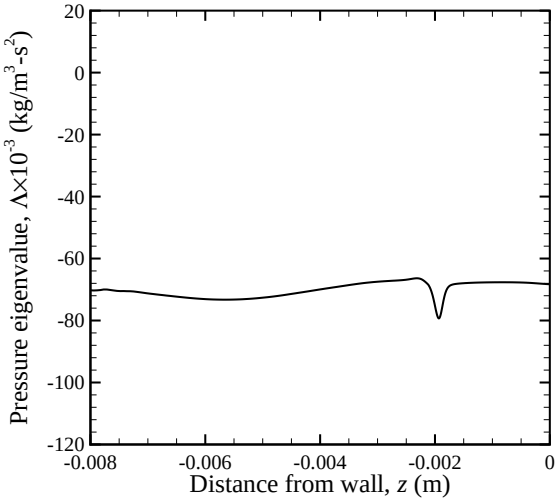
Figure C.12: Φ=0.90, Case B090-106



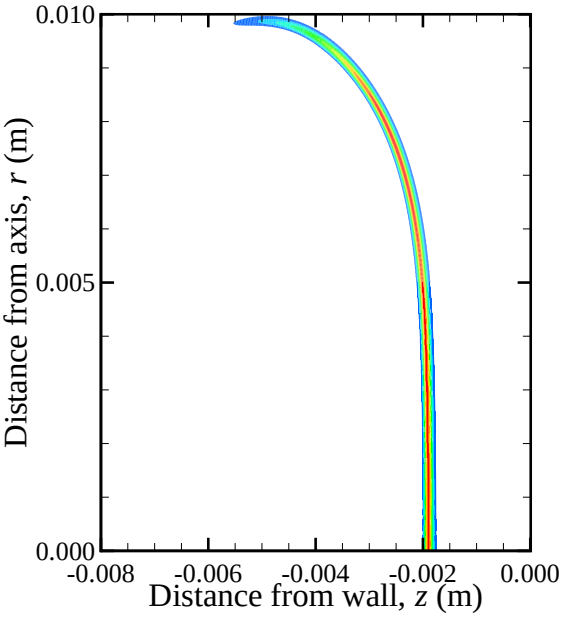
(a) Axial velocity profile



(b) CH mass fraction

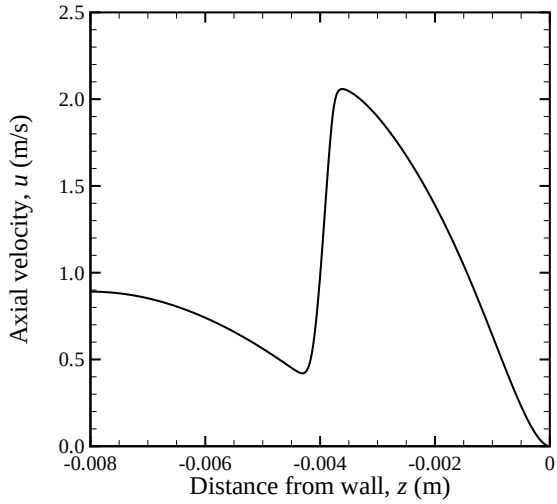


(c) Pressure eigenvalue

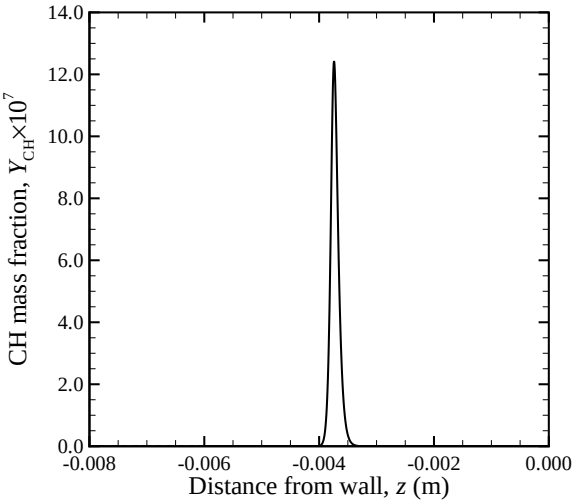


(d) CH contour

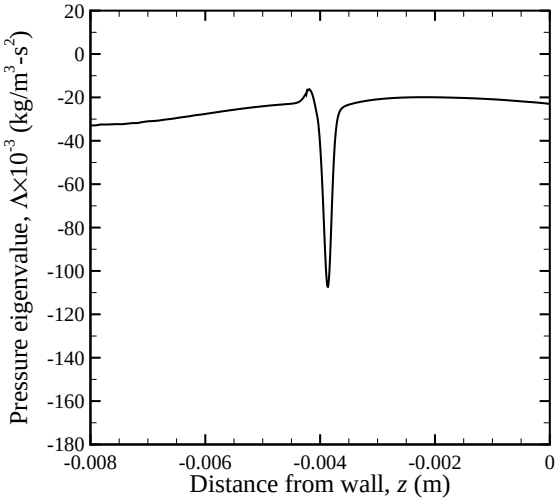
Figure C.13: Φ=0.90, Case B090-112



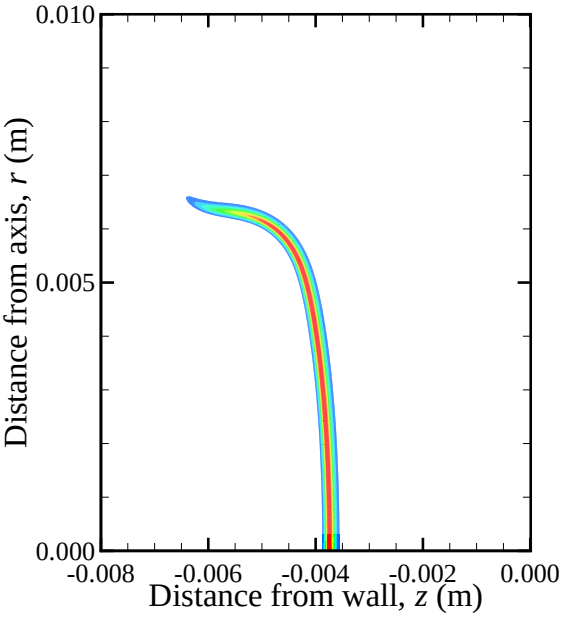
(a) Axial velocity profile



(b) CH mass fraction

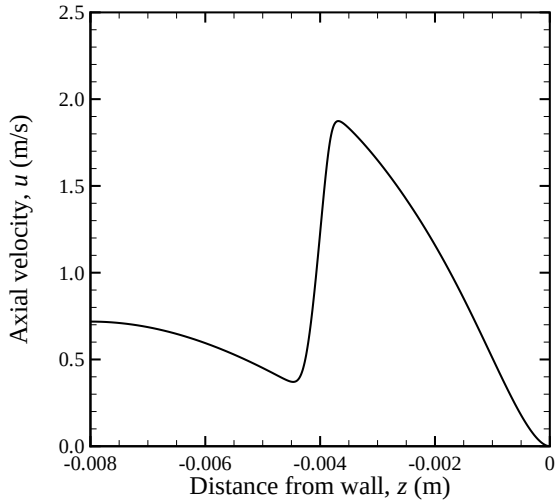


(c) Pressure eigenvalue

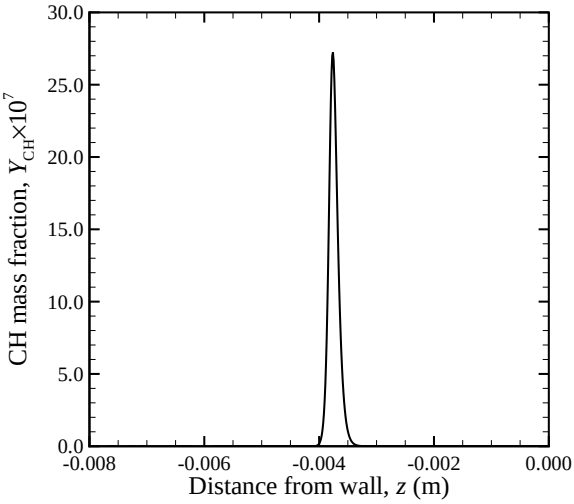


(d) CH contour

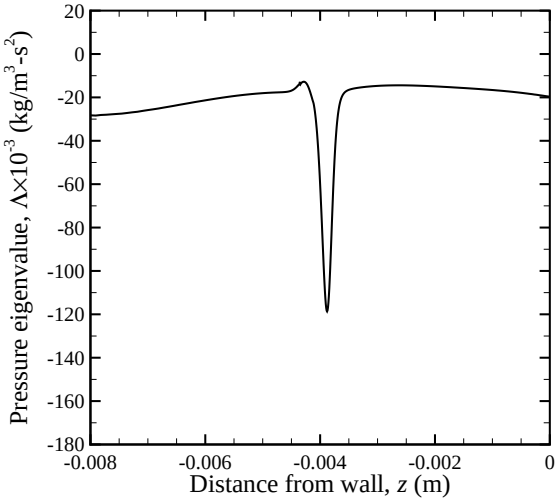
Figure C.14: Φ=1.00, Case B100-23



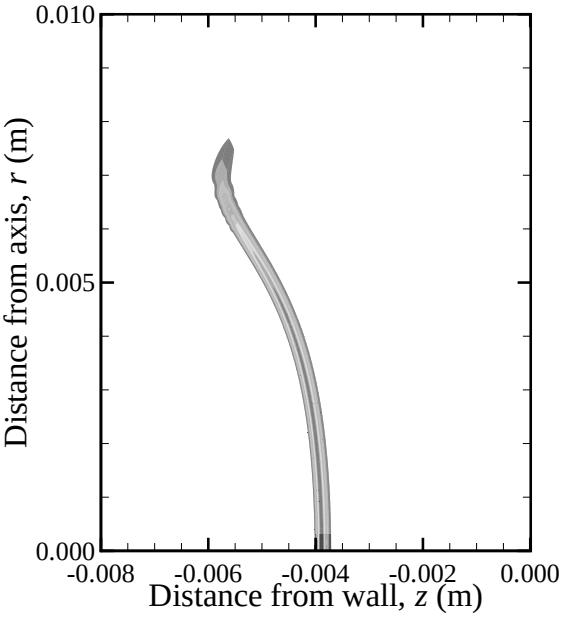
(a) Axial velocity profile



(b) CH mass fraction



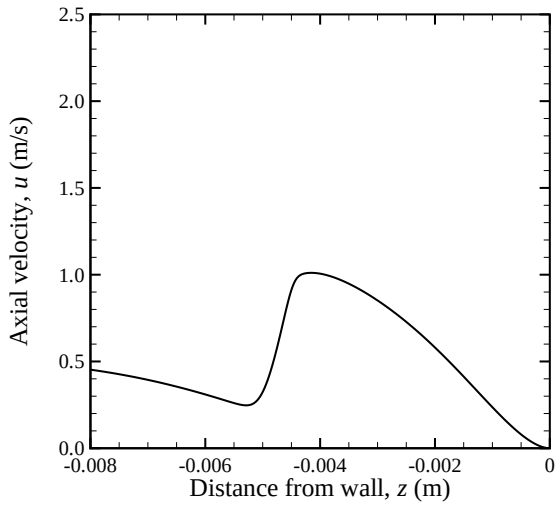
(c) Pressure eigenvalue



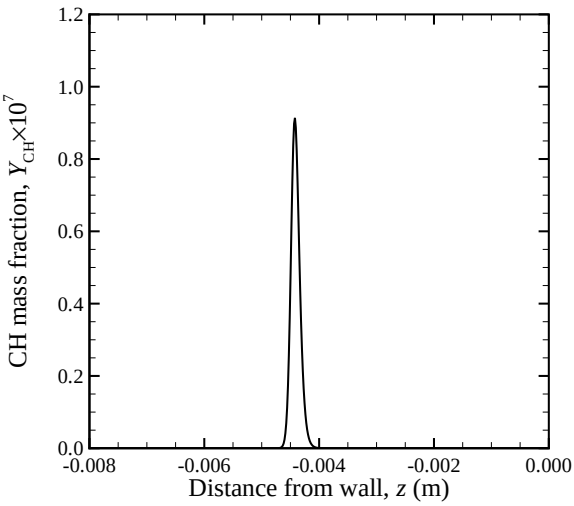
(d) CH contour

Figure C.15: Φ=1.20, Case B120-10

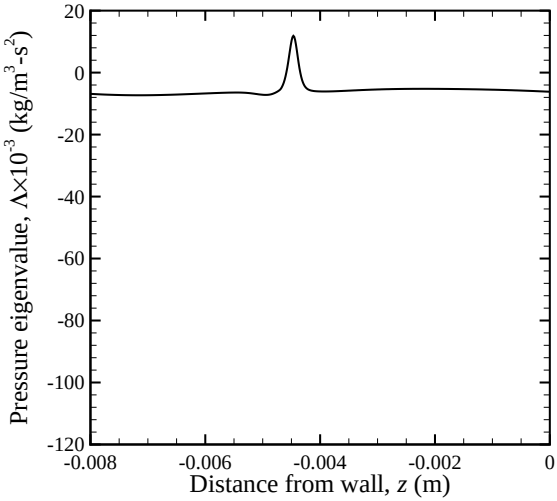
C.2 Phase III: 2D simulation of laboratory flames



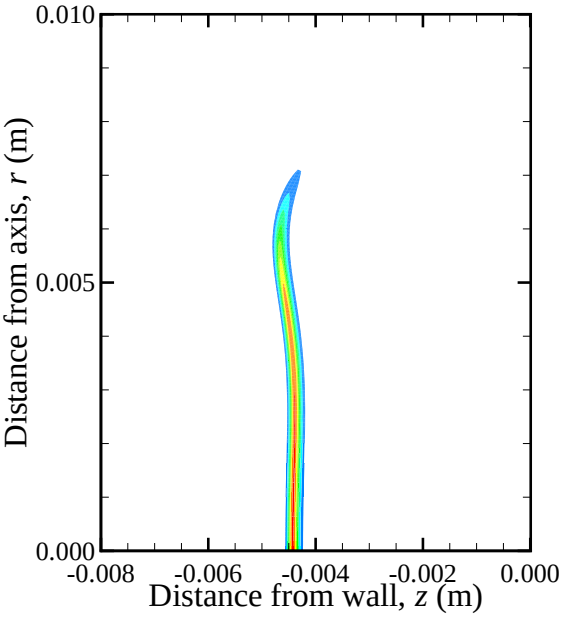
(a) Axial velocity profile



(b) CH mass fraction

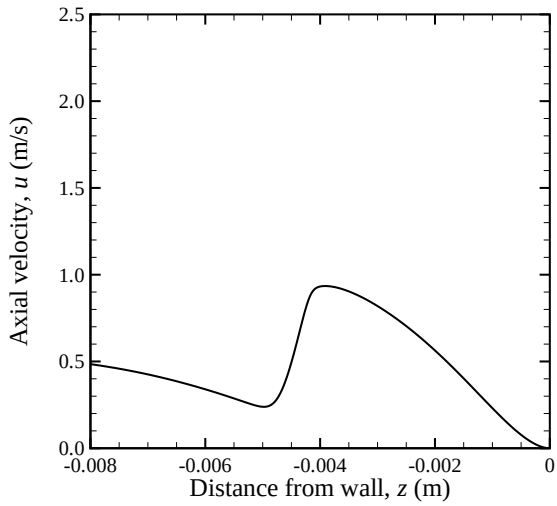


(c) Pressure eigenvalue

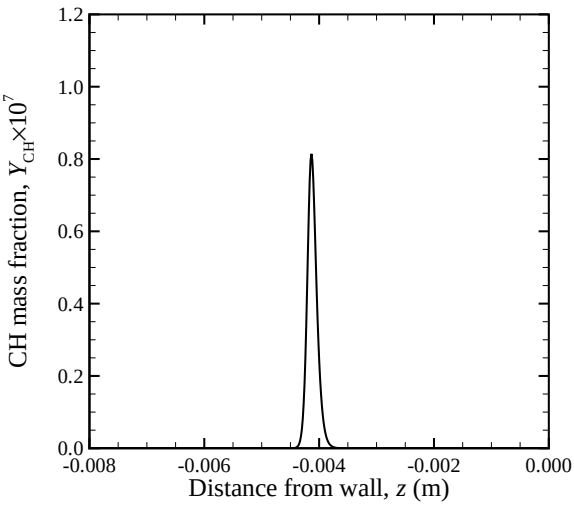


(d) CH contour

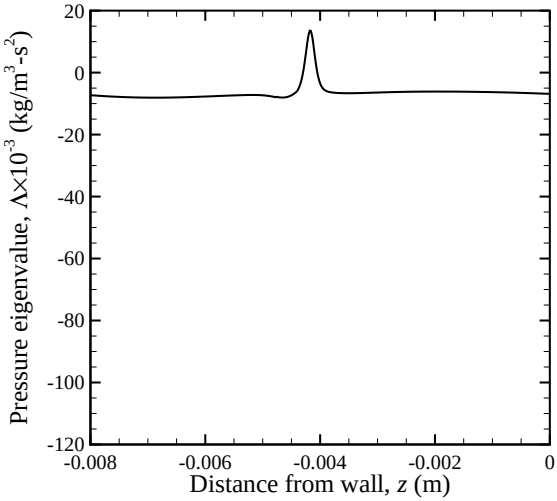
Figure C.16: Φ=0.70, Case C070-4



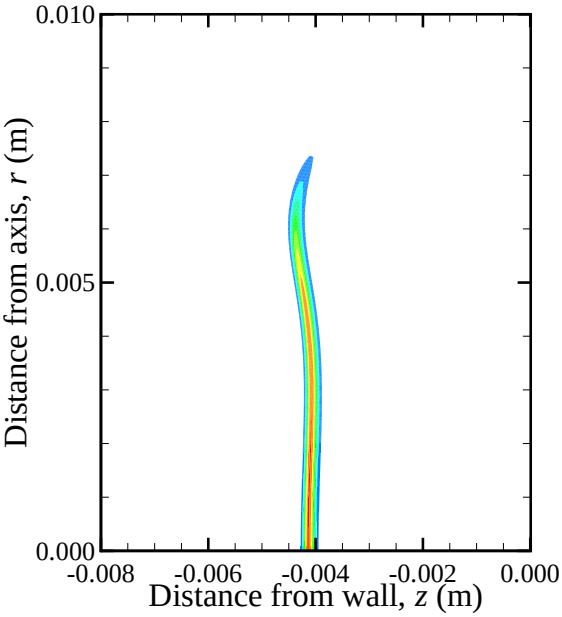
(a) Axial velocity profile



(b) CH mass fraction

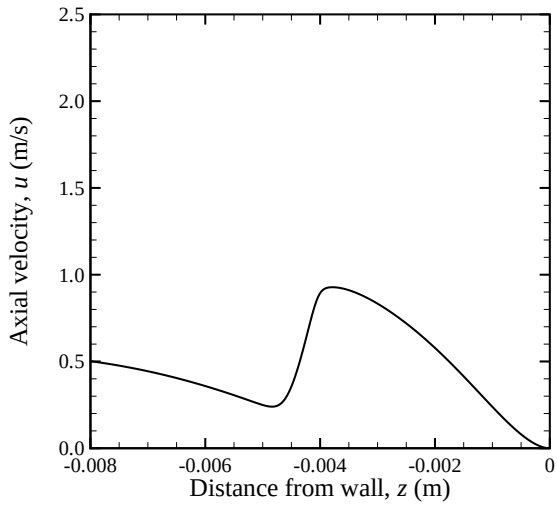


(c) Pressure eigenvalue

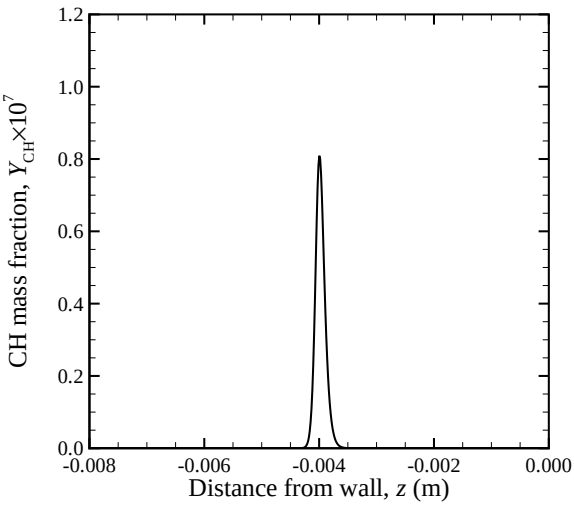


(d) CH contour

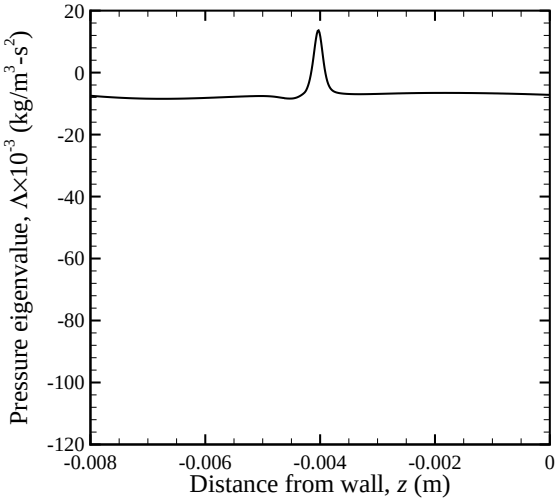
Figure C.17: Φ=0.70, Case C070-11



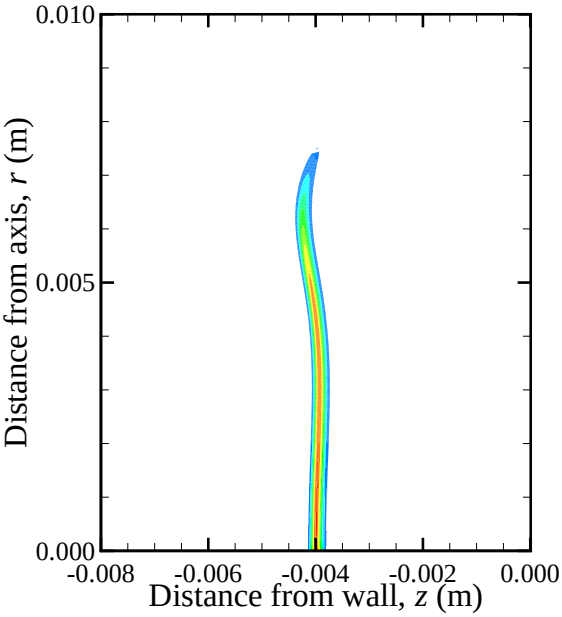
(a) Axial velocity profile



(b) CH mass fraction

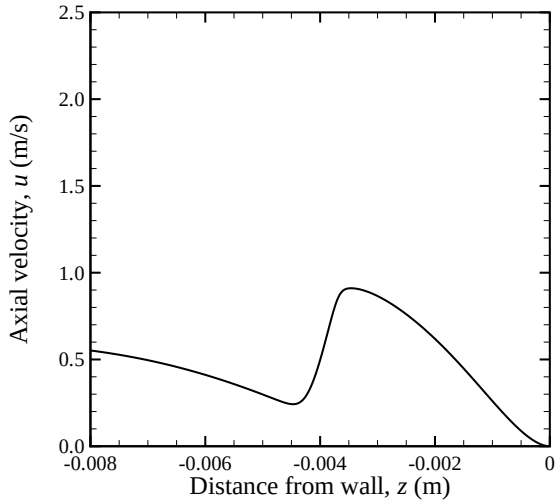


(c) Pressure eigenvalue

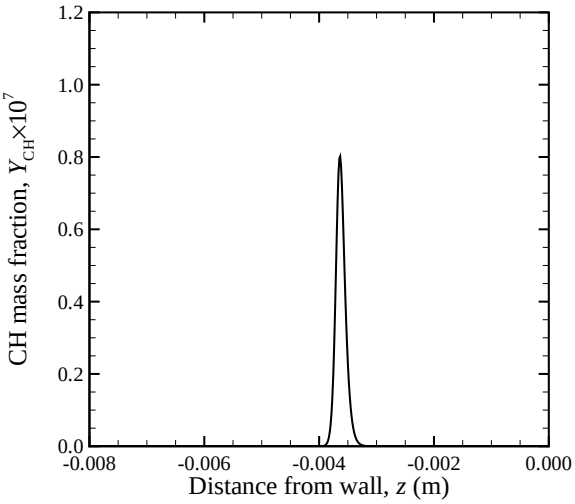


(d) CH contour

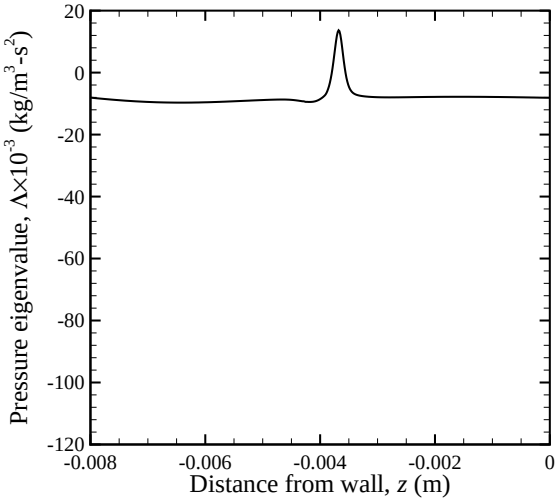
Figure C.18: Φ=0.70, Case C070-8



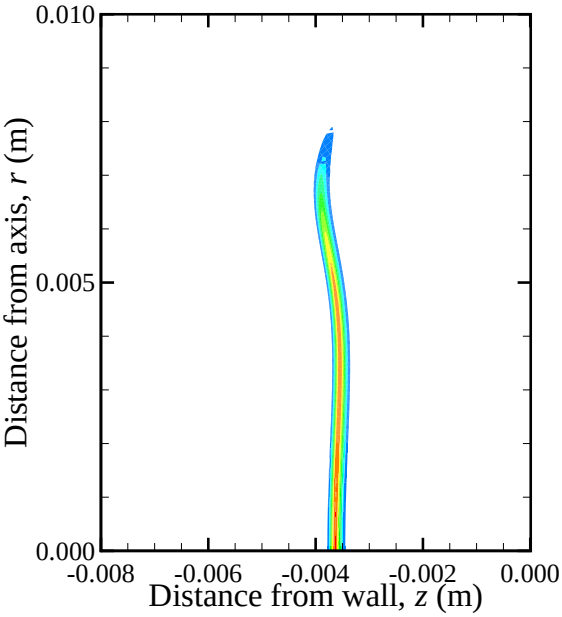
(a) Axial velocity profile



(b) CH mass fraction

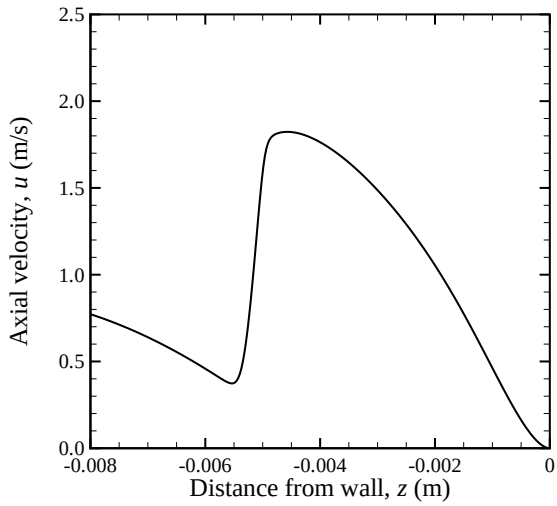


(c) Pressure eigenvalue

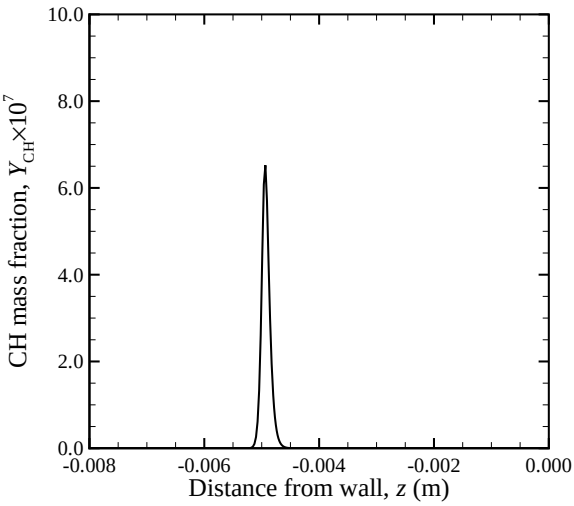


(d) CH contour

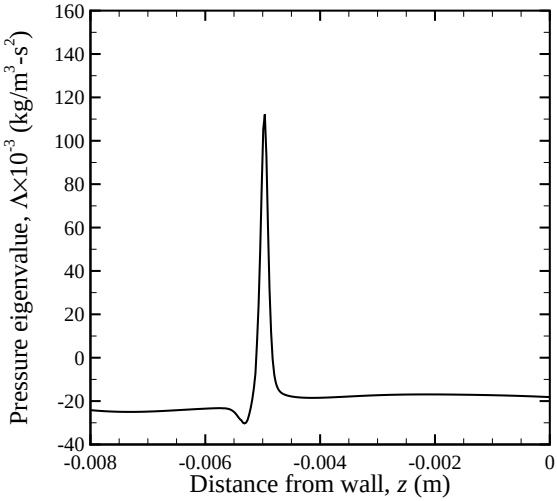
Figure C.19: Φ=0.70, Case C070-9



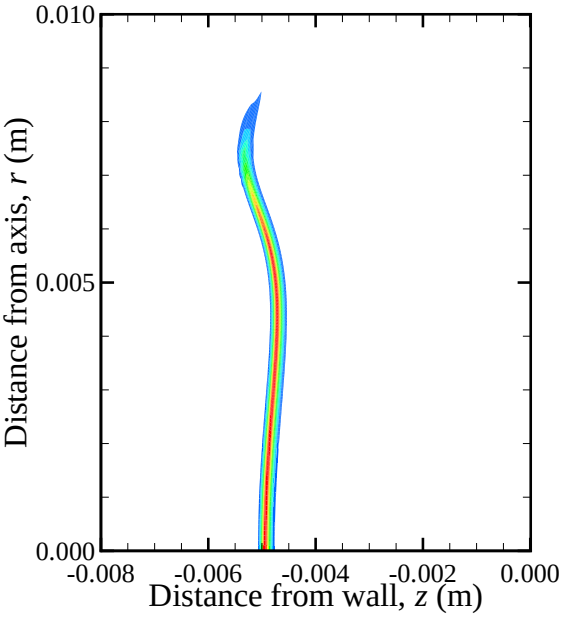
(a) Axial velocity profile



(b) CH mass fraction

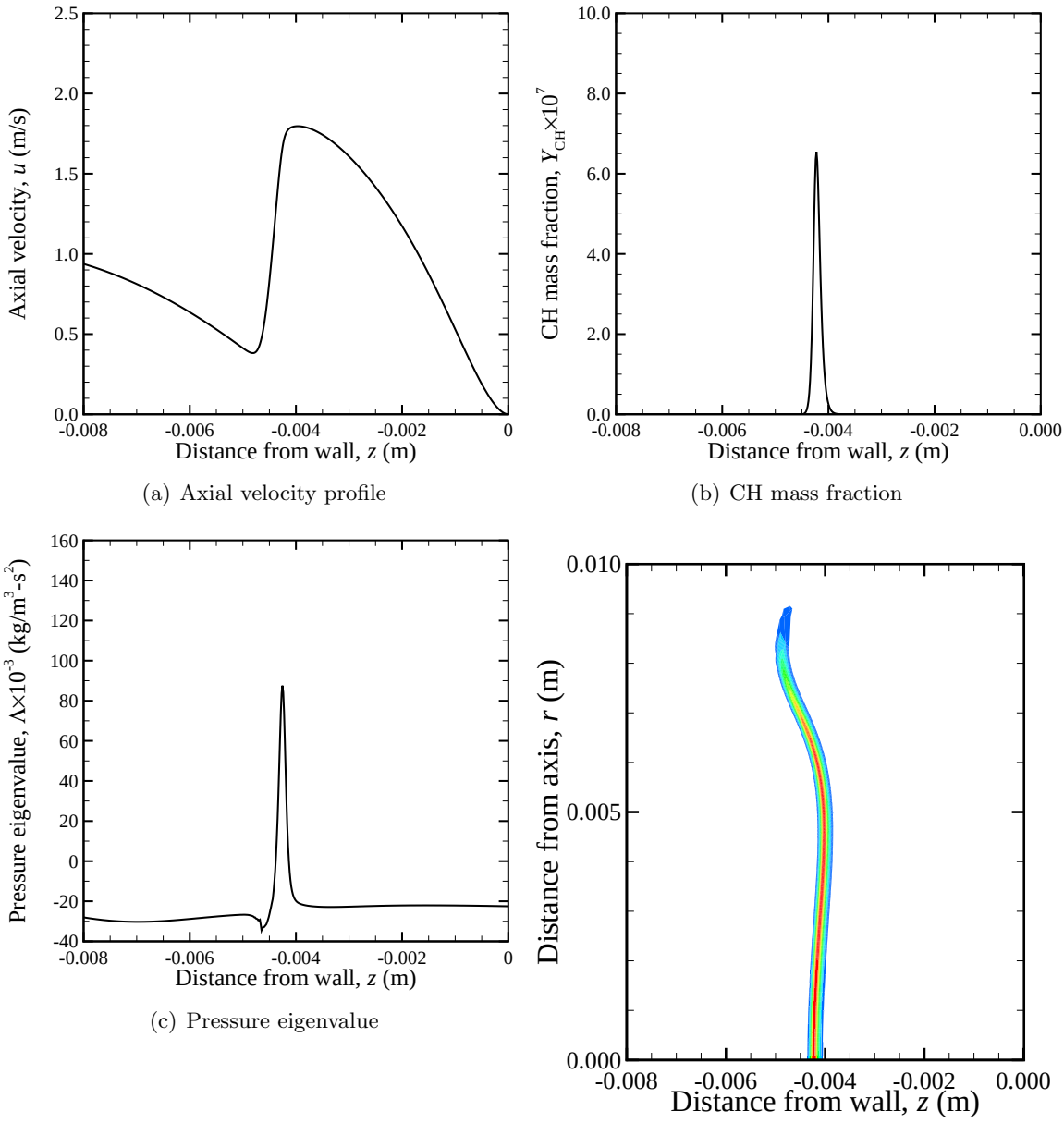


(c) Pressure eigenvalue



(d) CH contour

Figure C.20: Φ=0.90, Case C090-5



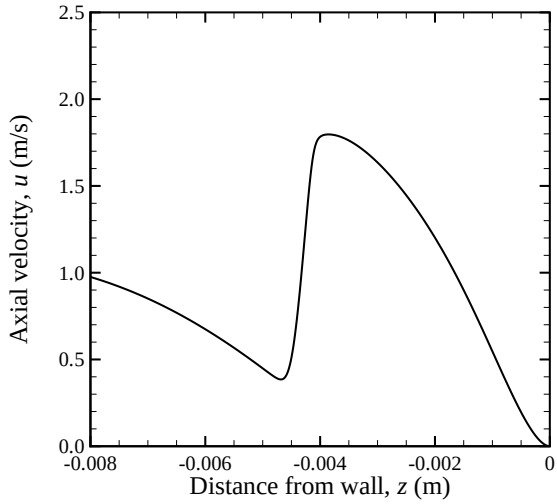
(a) Axial velocity profile

(b) CH mass fraction

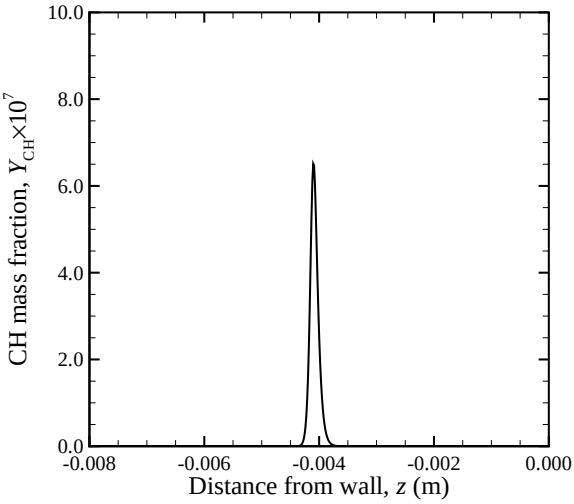
(c) Pressure eigenvalue

(d) CH contour

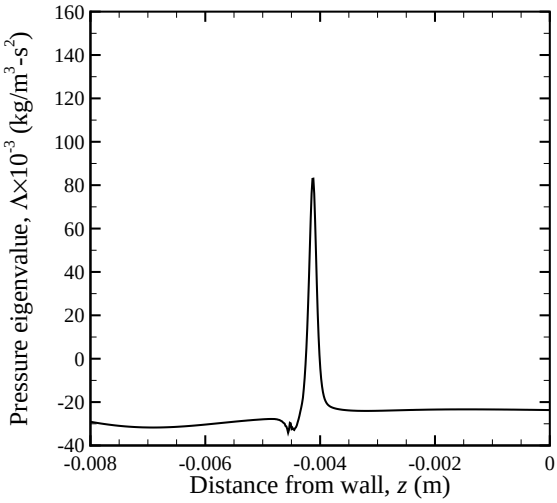
Figure C.21: Φ=0.90, Case C090-2



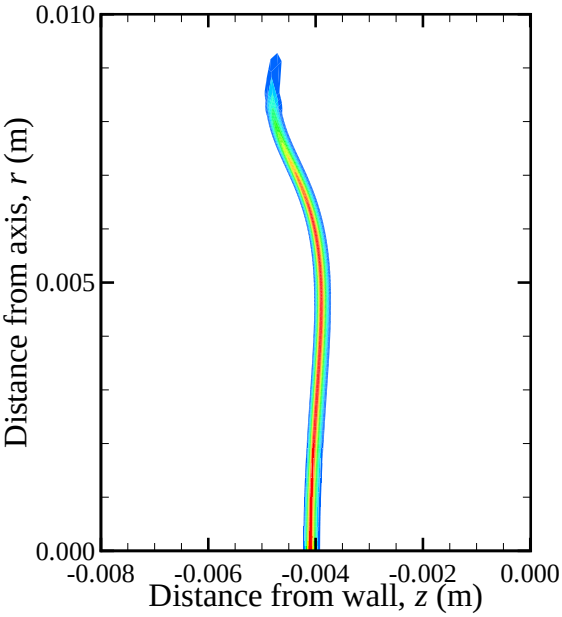
(a) Axial velocity profile



(b) CH mass fraction

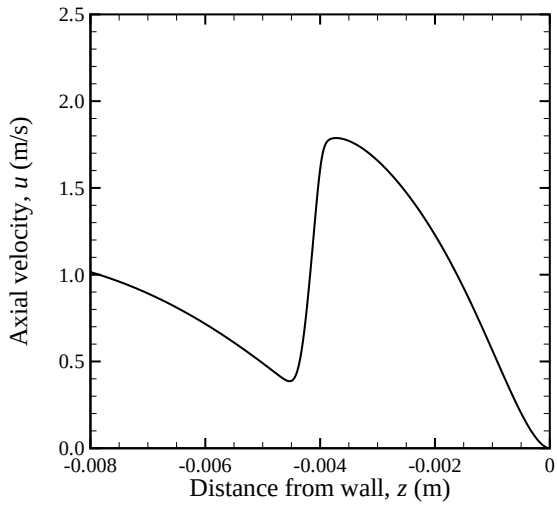


(c) Pressure eigenvalue

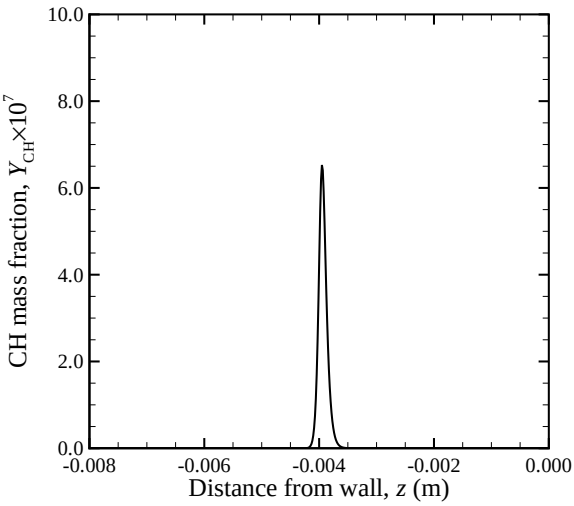


(d) CH contour

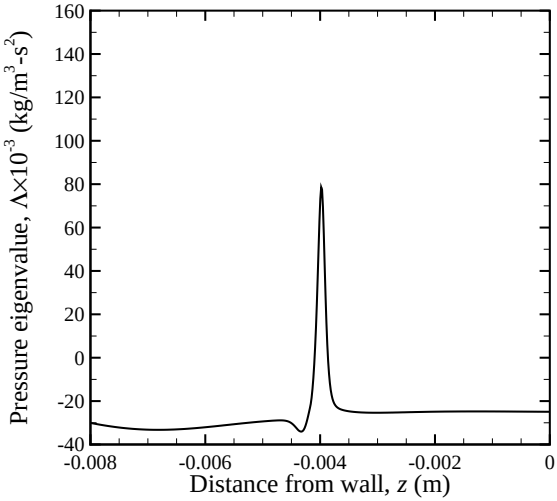
Figure C.22: Φ=0.90, Case C090-4



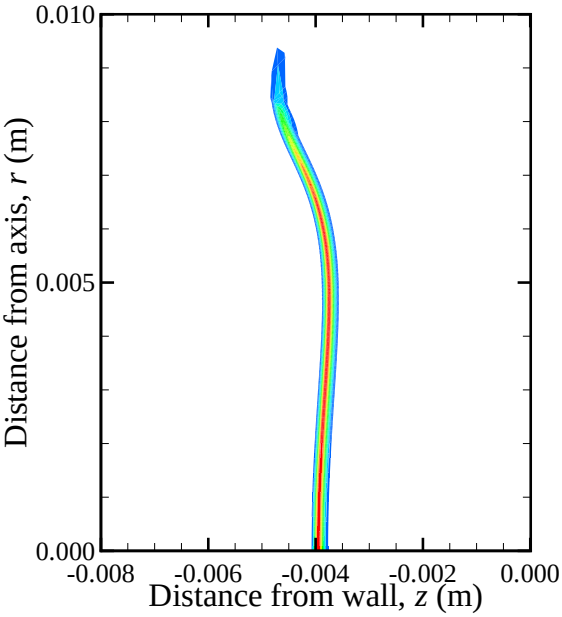
(a) Axial velocity profile



(b) CH mass fraction

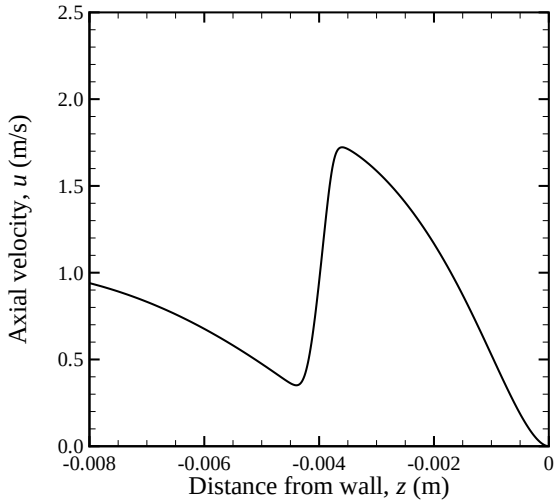


(c) Pressure eigenvalue

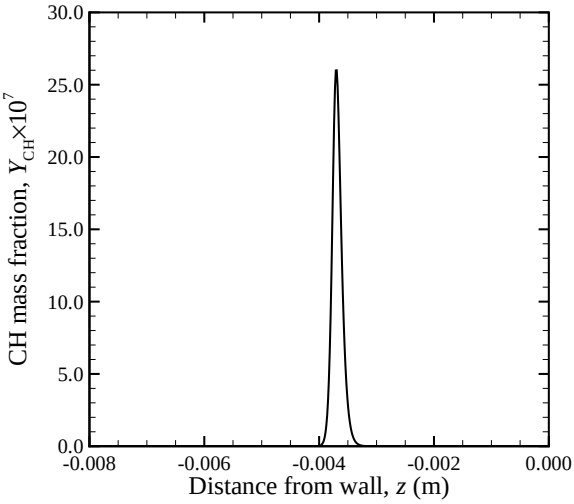


(d) CH contour

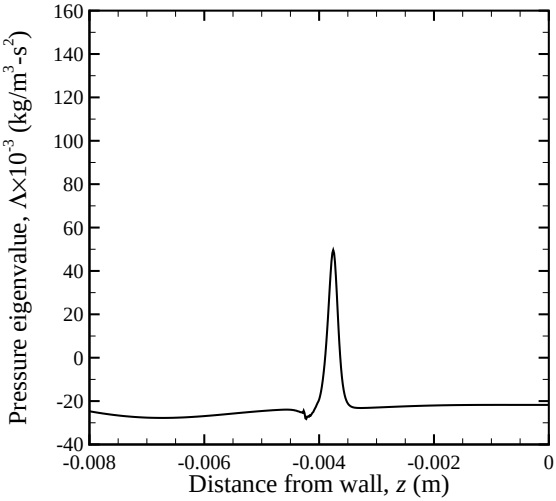
Figure C.23: Φ=0.90, Case C090-3



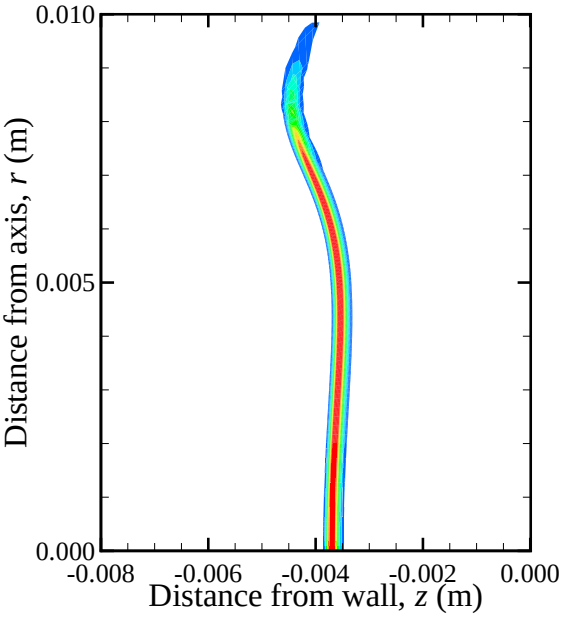
(a) Axial velocity profile



(b) CH mass fraction



(c) Pressure eigenvalue



(d) CH contour

Figure C.24: Φ=1.20, Case C120-1

Appendix D

Post-simulation analysis tool

D.1 PSV velocity simulation

The following *MATLAB*®script takes one-dimensional flow data and computes a particle motion. By filtering the particle motion as done in the laboratory, a simulation of PSV-measured velocity data can be obtained. For more details, Bergthorson (2005) should be consulted. The script was originally written by Jeff Bergthorson, with improvements made by the author and Laurent Benezech. It is included here for an archival purpose.

```
% PSV: PSV velocity correction.
% This program computes what PSV velocity data should be given simulation
% velocity data.
%
% USAGE: [PM, PSV, PFORCE, converged] = psv(data)
% returns the particle motion (PM) data and particle-streak velocimetry
% (PSV) data as well as three forces acting on a particle at each point.
% PM contains three columns.  The first column is the position of the
% particle.  The second is the velocity of the particle at that point and
% the third column is the acceleration of the particle at that point.  PSV
% contains two columns.  The first column is the position where the
% computed PSV velocity is defiend and the second column is the
% corresponding velocity data.  PFORCE contains three columns: Stokes drag,
% Thermophoretic force and gravity.  Each row corresponds to the first
% column of PM where particle locations are specified.  converged is a
```

```
% single integer value that returns 1 if the integration was successful and
% the particle reached to the end of the domain and 0 otherwise.  The
% function takes one to five input arguments.  The first input is the
% simulation data, which is mandatory and must be arranged in the following
% order: [x u V T r M mu] where the first column (x) is the axial position,
% u is the axial velocity, V is the spreading rate, dv/dr, T is the
% temperature, r is the density, M is the average molecular weight and mu
% is the mixture gas viscosity. It is assumed the data is stored such that
% it is an increasing order in x.  The second to sixth inputs are optional.
% The second input argument can by 'Al2O3', 'Zeeo' or user-specified
% particle properties.  When particle properties are specified, it must be
% 1x3 cell with the first cell being the particle diameter (m), the second
% being the particle density (Kg/m^3) and the last one is the thermal
% conductivity of the particle.  This can be a single number (constant
% thermal conductivity), or an array with the first column being
% temperature and the second column being the corresponding thermal
% conductivity at the given temperature.  (See example below)  When
% particle data is omitted, 'Zeeo' is used by default.  The third to sixth
% parameters are, the chopping frequency of the PSV, max iteration of
% temporal integration of particle motion, the number of subiterations
% within each PSV chopping cycle and the integration order (1 or 2).  When
% omitted, 1600, 100000, 100 and 2 are used by default respectively.
%
% example 1:
% psv(simdata, {1.0e-5, 1000, 1.0}): simulates a particle whose diameter is
% 10 microns, density is 1000 (kg/m^3), and the thermal conductivity is 1
% (W/m-K) at all temperature.
%
% example 2:
% psv(simdata, {3.0e-6, 2400, [300 100.0; 2000 1000.0]}: simulates a
% particle whose diameter is 3 microns, density is 2400 (kg/m^3), and the
% thermal conductivity is 100 (W/m-K) at 300K and 1000 (W/m-K) at 2000 K.
```

```
% At other temperature, the conductivity is linearly interpolated or
% extrapolated.
%
% example 3:
% psv(simdata, 'Al2O3', 2400, 250000, 200) : simulates a motion of Al2O3
% particle with PSV chopping frequency of 2400.  It tries to iterate up to
% a quater of million iterations at a time step of 1/2400 x 1/200.
%
% For more details, Jeff Bergthorson's thesis should be consulted.
%
% AUTHORS: Jeff M. Bergthorson, Laurent Benezech, and Kazuo Sone.
%
function [PM, PSV, PFORCE, converged] = psv(flowData, varargin)

narg = size(varargin,2);

if (nargin < 1)
   error('psv:NumInputs','Not enough input arguments.');
end

%%% Parameter definitions. %%%
if narg < 1
    % When particle is not specified, use 'Zeeo' as default.
    param = 'Zeeo';
else
    param = varargin{1};
end

% Particle properties, {diameter, density, conductivity}.

if iscell( param )
    % This is the case when property is specified by user.
```

```
    % dpart: Diameter of the particle.
    dpart = param{1};
    % rhopart: Particle density in kg/m^3
    rhopart = param{2};
    pcond = param{3};
else
    % Use preset values.
    if strcmp( param, 'Al2O3' )
        dpart = 1.0e-6;     % (m)
        rhopart = 3830.0;   % (Kg/m^3)
        %Thermal conductivity of particles Incropera and deWitt
        % - Fundamentals of Heat and Mass Transfer, 3rd edition 1990
        pcond = [200 55;
            400 26.4;
            600 15.8;
            800 10.4;
            1000 7.85;
            1200 6.55;
            1500 5.66;
            2000 6;
            2500 6.4];
    elseif strcmp( param, 'Zeeo' )
        dpart = 3.0e-6;     % (m)
        rhopart = 2400.0;   %(kg/m^3)
        pcond = 2.3;        % (W/mK)
    else
        % Error!
        error('psv:ParticleInfo','Particle specification incorrect.');
    end
end

% frequency: Chopping frequency of PSV.
```

```
if narg > 1
    frequency = varargin{2};
else
    frequency = 1600;
end

% max_iterations: Maximum number of integration steps.
if narg > 2
    max_iterations = varargin{3};
else
    max_iterations=100000;
end

% nsubcycle: number of iterations to get 1 PSV cycle.
% Default value is 100.
if narg > 3
    nsubcycle = uint16( varargin{4} );
else
    nsubcycle = 100;
end

% integorder: Order of time integration (1 or 2).
% Default value is 2.
% Use 1st order only when speed is important.
if narg > 4
    integorder = uint16( varargin{5} );
    if integorder < 1 || integorder > 2
        error('psv:IntegOrder', ...
              'Incorrect integration order.  Must be 1 or 2');
    end
else
    integorder = 2;
```

```
end

% deltat: Integration of particle equations.
deltat = 1/frequency * 1/double(nsubcycle);
% halfdeltat: 1/2 deltat (used in 2nd order method)
halfdeltat = 0.5 * deltat;

%%% Universal constants. %%%
% Using the simulated flow, temperature and property fields,
% we need to model a particles behavior as it traverses the flowfield.
%Gravitational acceleration
a_gravity=9.81; %m/s^2

% mpart: Mass of the particle
mpart=rhopart*pi*4/3*(dpart/2)^3; %in kg;
%
% End of parameters.
%

xsim=flowData(:,1);
lsim=max(xsim); % End of the simulation domain.

usim=flowData(:,2);
Vsim=flowData(:,3);
T=flowData(:,4);
rho=flowData(:,5);
Mbar=flowData(:,6);
visc=flowData(:,7);

gradT=zeros(size(flowData,1),1);
for i=2:1:size(flowData,1)-1;
    gradT(i)=(T(i+1)-T(i-1))/(xsim(i+1)-xsim(i-1));
```

```
end
gradT(size(flowData,1))=gradT(size(flowData,1)-1);

%
%Initialize the particle with a corrected particle velocity and
%acceleration
i = 1;
tpart(1)=0;
xpart(1)=0;

% CKWp: approximately 1+1.142*Kn for small Kn.
%     : see Eqn.(A.10) in Jeff Bergthorson's thesis.
CKWp(1)=1.05;

% vpart: Eqns.(A.23) & (A.24) in Jeff Bergthorson's thesis
%      : vpart/vfluid = 1/(1+C_KW*tau_S*sigma_f)
%      : sigma_f = du/dx.
dudx = -2*Vsim(1);
vpart(1)=usim(1)/(1+CKWp(1)*(rhopart*dpart^2/(18*visc(1)))*dudx);
% ufp: fluid velocity at particle location.
ufp(1)=usim(1);
[apart(1), F_thermophoretic(1), F_stokes(1)] = ...
    Particle_acceleration(mpart, dpart, pcond, vpart(1), ...
                          [ufp(1) T(1) gradT(1) rho(1) Mbar(1) visc(1)]);

%The gravitational force is a constant and can be calculated prior to the
%loop
F_gravity=-mpart*a_gravity;

continue_iteration = 1;
while continue_iteration
    % Increment time by deltat, and update x, u,and a through 1st order
```

```
% explicit Euler method of integration.
i = i + 1;
tpart(i)=tpart(i-1)+deltat;
xpart(i)=xpart(i-1)+deltat*vpart(i-1);
vpart(i)=vpart(i-1)+deltat*apart(i-1);

% Interpolate the simulated fields at the particle location.
flow_at_particle = interp1(xsim, [usim T gradT rho Mbar visc], ...
                           xpart(i), [], 'extrap');

if integorder == 2
    apart(i) = Particle_acceleration(mpart, dpart, pcond, vpart(i), ...
                                     flow_at_particle);

    % Apply second order correction (2nd order Runge-Kutta method)
    xpart(i) = xpart(i-1) + halfdeltat*(vpart(i-1)+vpart(i));
    vpart(i) = vpart(i-1) + halfdeltat*(apart(i-1)+apart(i));

    % Interpolate the simulated fields at the particle location.
    flow_at_particle = interp1(xsim, [usim T gradT rho Mbar visc], ...
                               xpart(i), [], 'extrap');
end

[apart(i), F_thermophoretic(i), F_stokes(i)] = ...
    Particle_acceleration(mpart, dpart, pcond, vpart(i), ...
                          flow_at_particle);

% Check if the particle has reached to the wall close enough.
% and if so, exit the loop.
% Note xpart has been computed way above here, but I have to get apart
% otherwise the resulting array does not have the same number of rows.
if xpart(i) >= lsim || i > max_iterations
```

```
        continue_iteration = 0;
    end
end

% Issue an warning message if not converged.
if i > max_iterations
    converged = 0;
    warning('Iteration may not have converged.  ');
    warning('Increase maximum number of iterations.');
else
    converged = 1;
end

%now want to estimate the PSV velocity from this flowfield.
nb0 = nsubcycle+1;
nb1 = size(tpart,2);
ne0 = 1;
ne1 = size(tpart,2)-nsubcycle;
PSV(:,1) = (xpart(nb0:nb1)+xpart(ne0:ne1)) * 0.5;
PSV(:,2) = (xpart(nb0:nb1)-xpart(ne0:ne1)) * frequency;

% Assign to the returning matrix.
PM = [xpart; vpart; apart]';
PFORCE = [F_stokes; F_thermophoretic; F_gravity*ones(1,i)]';

%-------------------------
% Private functions
%-------------------------

%--------------------------------------------------------------------------
% Particle_acceleration: This function computes a particle acceleration.
function [apart, F_thermophoretic, F_stokes] = ...
```

```
    Particle_acceleration(mpart, dpart, pcond, vpart, flow_p)

ufp = flow_p(1);
Tp = flow_p(2);
gradTp = flow_p(3);
rhop = flow_p(4);
Mbarp = flow_p(5);
viscp = flow_p(6);

% R_univ: Universal gas constant.
R_univ=8314; %J/kmolK

% kftransp: The tranlational part of the thermal conductivity
%         : See Eqn.(A.26) in Jeff's thesis.
kftransp = 15*R_univ*viscp/(4*Mbarp);

% kpp : Thermal conductivity of the particle.  When it is not a function of
% temperature, use it.  If it is a function of temperature, pcond should be
% given by a table - simply interpolate.
if length(pcond) == 1
    kpp = pcond;
else
    kpp = interp1(pcond(:,1), pcond(:,2), Tp, [], 'extrap');
end

% Now calculate the parameters needed for the forces:
% Knp: Knudsen number.
Knp = Knudsen_number(Tp, Mbarp, viscp, rhop, dpart, R_univ);

% CKWp: Eqn.(A.10)
CKWp = 1+Knp*[1.142+0.558*exp(-0.999/Knp)];
```

```
%Now calculate the thermophoretic and stokes drag forces:
F_thermophoretic = Force_Thermophoretic_Talbot(viscp, dpart, rhop, ...
                                                kftransp, kpp, Knp, gradTp, Tp);
F_stokes = Force_Stokes(viscp, dpart, vpart, ufp, CKWp);

%Calculate the acceleration: Eqn.(A.7)
apart = (F_stokes+F_thermophoretic)/mpart;
%-------------------------------------------------------------------------

%-------------------------------------------------------------------------
% Force_Stokes: This function computes stokes drag.
function [F_stokes] = Force_Stokes(viscp, dpart, vpart, ufp, CKWp)

%Stokes drag: Eqn.(A.9)
F_stokes = -3*pi*viscp*dpart*(vpart-ufp)/CKWp;
%-------------------------------------------------------------------------

%-------------------------------------------------------------------------
% Force_Thermophoretic_Talbot: This function computes thermophoretic force
% by Talbot's model.
function [F_tp] = Force_Thermophoretic_Talbot(viscp, dpart, rhop, ...
                                                kftransp, kpp, Knp,gradTp, Tp)

%Talbot: Eqn.(A.28)
num = -(6*pi*viscp^2/rhop*dpart*1.17*(kftransp/kpp+2.18*Knp)*gradTp/Tp);
den = (1+3*1.14*Knp)*(1+2*kftransp/kpp+2*2.18*Knp);
F_tp = num/den;
%-------------------------------------------------------------------------

%-------------------------------------------------------------------------
% Knudsen_number: This function computes Knudsen number
function [Knp] = Knudsen_number(Tp, Mbarp, viscp, rhop, dpart, R_univ)
```

```
% cbarp: Eqn.(A.12)
cbarp = (8*R_univ*Tp/(Mbarp*pi))^0.5;

% Lambdap: Eqn.(A.11)
Lambdap=2*viscp/(rhop*cbarp);

% Knp: Knudsen number.
Knp = 2*Lambdap/dpart;
%-------------------------------------------------------------------------
```

Appendix E

Opposed-jet hydrogen flames and comparison of mechanisms

This appendix is designed to demonstrate the importance of the hydrogen subset of hydrocarbon mechanisms. Understanding hydrogen flame behavior should supplement hydrocarbon combustion models because each hydrocarbon reaction model contains a hydrogen reaction submechanism (see Table 1.1). Figs. E.1–E.4 compare performance of three different hydrogen mechanisms: H1, H2, and H4 in Table 1.1, in the above opposed-jet configuration at the same condition. H2 and H4 tend to show an agreement, in particular, their velocity profile is nearly identical, while the GRI-Mech 3.0 submechanism shows discernible departure from the other two. However, they all disagree in the prediction of hydrogen radical profile. Some of the reactions that appear in the hydrogen reaction submodel have a significant effect on the flame speed of hydrocarbons, as pointed out by several authors (Williams, 2000; Turányi *et al.*, 2002), and refinement in the hydrocarbon kinetics model should come in tandem with understanding of hydrogen reactions, including extinction characteristics.

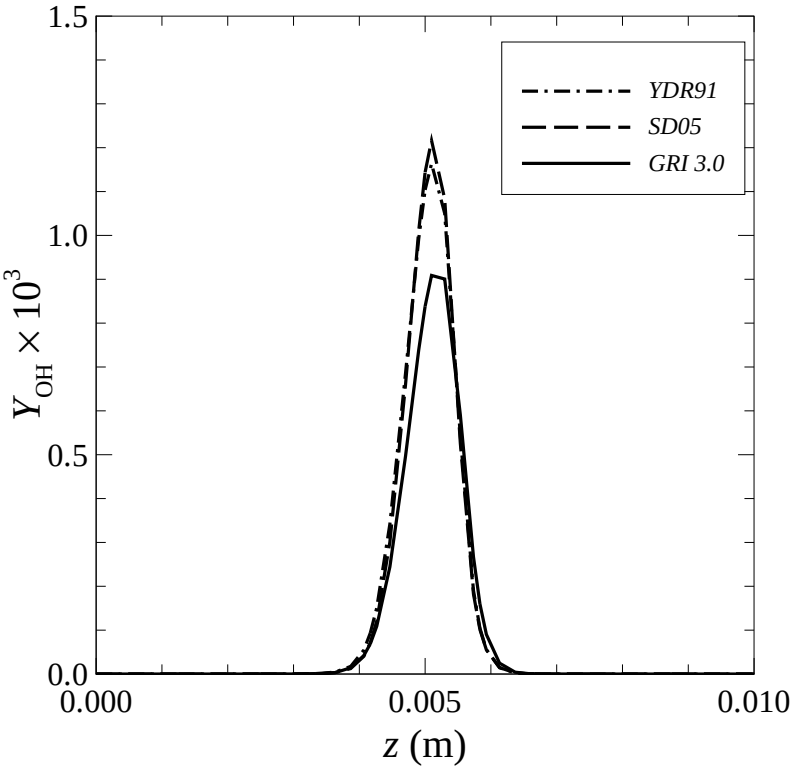


Figure E.1: Comparison of hydroxyl radical mass fraction along the axis between three different hydrogen mechanisms: GRI-Mech 3.0 [H1] (solid line), SD05 [H2] (dashed line), and YDR91 [H4] (dot-dashed line)

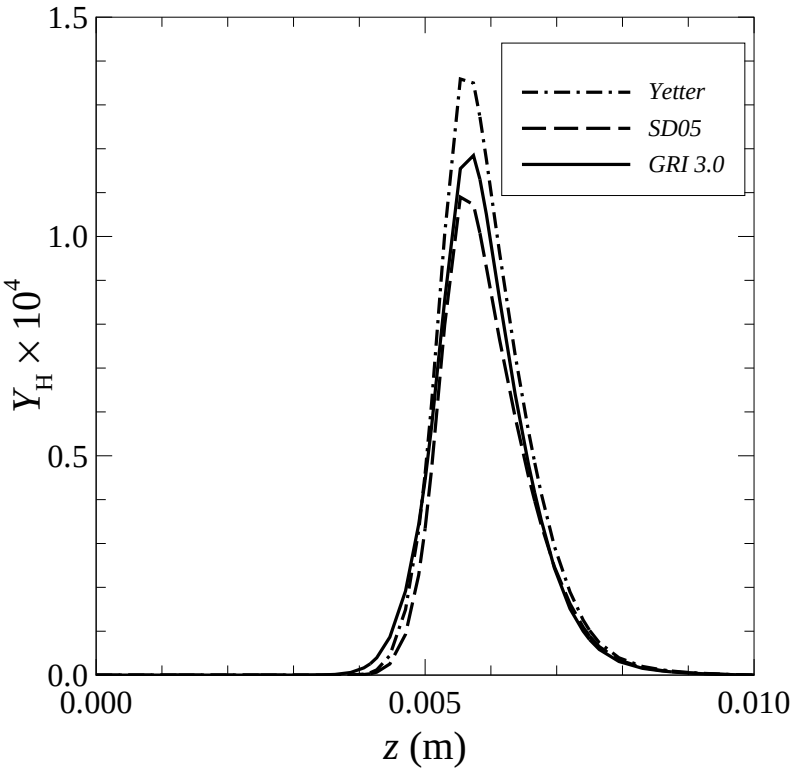


Figure E.2: Comparison of hydrogen radical mass fraction along the axis between three different hydrogen mechanisms. (Legend as in Fig. E.1.)

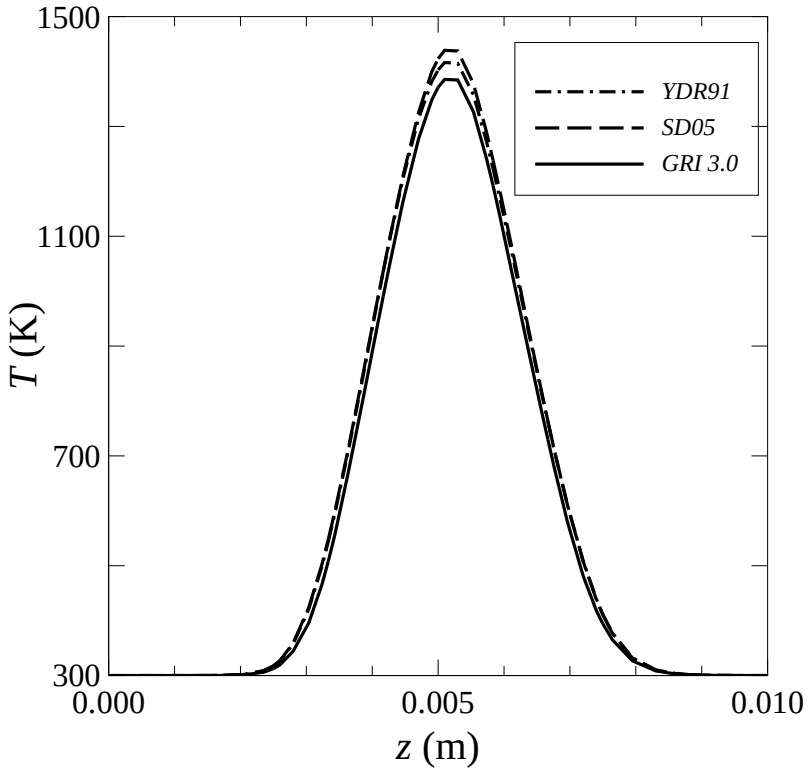


Figure E.3: Comparison of temperature profile along the axis between three different hydrogen mechanisms. (Legend as in Fig. E.1.)

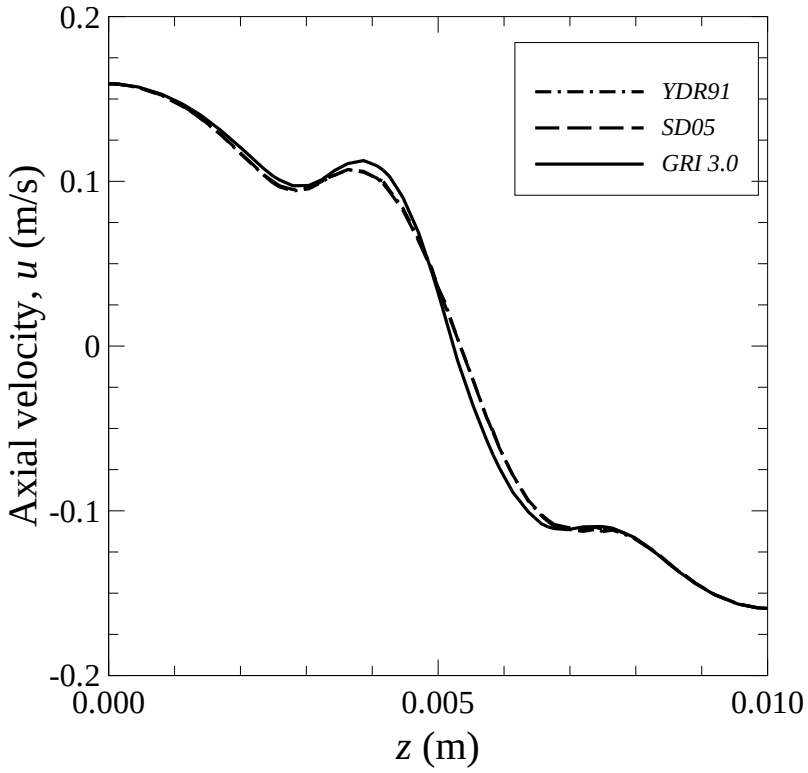


Figure E.4: Comparison of axial velocity profile along the axis between three different hydrogen mechanisms. (Legend as in Fig. E.1.)

Bibliography

Aivazis, M. A. G., April 7-10 2002. Fuego: an extensible toolkit for building chemical kinetics and thermodynamics applications. Presented at Ninth International Conference on Numerical Combustion, Sorrento, Italy. `http://web.ing.unisannio.it/icnc2002/`.

Amsden, A. A., March 1993. KIVA-3: A KIVA Program with Block-Structured Mesh for Complex Geometries. Technical Report LA–12503–MS, Los Alamos National Laboratory.

Amsden, A. A., July 1997. KIVA-3V: A Block-Structured KIVA Program for Engines with Vertical or Canted Valves. Technical Report LA–13313–MS, Los Alamos National Laboratory.

Amsden, A. A., May 1999. KIVA-3V, Release 2, Improvements to KIVA-3V. Technical Report LA–113608–MS, Los Alamos National Laboratory.

Amsden, A. A., O'Rourke, P. J., & Butler, T. D., May 1989. KIVA-II: A Computer Program for Chemically Reactive Flows with Sprays. Technical Report LA–11560–MS, Los Alamos National Laboratory.

Amsden, A. A., Ramshaw, J. D., O'Rourke, P. J., & Dukowicz, J. K., February 1985. KIVA: A Computer Program for Two- and Three-Dimensional Fluid Flows with Chemical Reactions and Fuel Sprays. Technical Report LA–10245–MS, Los Alamos National Laboratory.

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Croz, J. D., Greenbaum, A., Hammarling, S., McKenney, A., & Sorensen, D., 1999. *LAPACK Users' Guide*. SIAM, third edition.

Beizer, B., 1990. *Software Testing Techniques*. International Thomson Computer Press, second edition.

Bell, J. B., Day, M. S., Grcar, J. F., & Lijewski, M. J., 2005a. Stochastic algorithms for the analysis of numerical flame simulations. *Journal of Computational Physics*, **202**: 262–280.

Bell, J. B., Day, M. S., Shepherd, I. G., Johnson, M. R., Cheng, R. K., Grcar, J. F., Beckner, V. E., & Lijews, M. J., 2005b. Numerical simulation of a laboratory-scale turbulent v-flame. *Proceedings of the National Academy of Science of the United States of America*, **102**(29):10006–10011.

Benezech, L. J., Bergthorson, J. M., & Dimotakis, P. E., 2006. Experimental investigation of planar strained methane-air and ethylene-air flames. Technical Report GALCITFM:2006.002, California Institute of Technology. `http://resolver.caltech.edu/CaltechGALCITFM:2006.002`.

Bennett, B. A. V., Fielding, J., Mauro, R. J., Long, M. B., & Smooke, M. D., 1999. A comparison of the structures of lean and rich axisymmetric laminar bunsen flames: application of local rectangular refinement solution-adaptive gridding. *Combustion Theory and Modelling*, **3**(4):657–687.

Bergthorson, J. M., May 2005. *Experiments and Modeling of Impinging Jets and Premixed Hydrocarbon Stagnation Flames*. Ph.D. thesis, California Institute of Technology, Pasadena, CA. `http://resolver.caltech.edu/CaltechETD:etd-05242005-165713`.

Bergthorson, J. M., Sone, K., Mattner, T. W., Dimotakis, P. E., Goodwin, D. G., & Meiron, D. I., 2005a. Impinging laminar jets at moderate reynolds numbers and separation distances. *Physical Review E*, **72**:066307.

Bergthorson, J. M. & Dimotakis, P. E., 2007. Premixed laminar c_1c_2 stagnation flames: Experiments and simulations with detailed thermochemistry models. *Proceedings of the Combustion Institute*, **31**:1139–1147.

Bergthorson, J. M., Goodwin, D. G., & Dimotakis, P. E., 2005b. Particle streak velocimetry and CH laser-induced fluorescence diagnostics in strained, premixed, methaneair flames. *Proceedings of the Combustion Institute*, **30**:1637–1644.

Bird, R. B., Stewart, W. E., & Lightfoot, E. N., 2002. *Transport Phenomena*. J. Wiley, New York.

Blackburn, H. M. & Henderson, R. D., 1999. A study of two-dimensional flow past an oscillating cylinder. *Journal of Fluid Mechanics*, **385**:255–286.

Blackburn, H. M. & Lopez, J. M., 2002. Modulated rotating waves in an enclosed swirling flow. *Journal of Fluid Mechanics*, **465**:33–58.

Blackburn, H. M. & Sherwin, S. J., 2004. Formulation of a Galerkin spectral element-Fourier method for three-dimensional incompressible flows in cylindrical geometries. *Journal of Computational Physics*, **197**:759–778.

Blackford, L. S., Choi, J., Cleary, A., D'Azevedo, E., Demmel, J., Dhillon, I., Dongarra, J., Hammarling, S., Henry, G., Petitet, A., Stanley, K., Walker, D., & Whaley, R. C., 1997. *ScaLAPACK Users' Guide*. SIAM.

Boyd, J. P., 2000. *Chebyshev and Fourier spectral methods*. Dover Publications, Inc., Mineola, NY, second edition.

Bradley, D., Gaskell, P. H., & Gu, X. J., 1996. Burning velocities, markstein lengths, and flame quenching for spherical methane-air flames: A computational study. *Combustion and Flame*, **104**:176–198.

Buckmaster, J. D. & Ludford, G. S. S., 1982. *Theory of laminar flames*. Cambridge University Press.

Candel, S. M. & Poinsot, T. J., 1990. Flame stretch and the balance equation for the flame area. *Combustion Science and Technology*, **70**:1–15.

Celik, I., Yavuz, I., Smirnov, A., Smith, J., Amin, E., & Gel, A., 2000. Prediction of in-cylinder turbulence for IC engines. *Combustion Science and Technology*, **153**:339–368.

Chung, S. H. & Law, C. K., 1984. An invariant derivation of flame stretch. *Combustion and Flame*, **55**:123–125.

Clavin, P. & Joulin, G., 1983. Premixed flames in large scale and high intensity turbulent flow. *Le Journal De Physique—Letters*, **44**:L–1–L–12.

Coffee, T. & Heimerl, J., 1981. Transport algorithms for premixed, laminar steady-state flames. *Combustion and Flame*, **43**:273–289.

Cohen, S. D. & Hindmarsh, A. C., September 1994. CVODE user guide. Technical Report LLNL Report UCRL-MA-118618, Lawrence Livermore National Laboratory.

Curtiss, C. F. & Bird, R. B., 1999. Multicomponent diffusion. *Industrial & Engineering Chemistry Research*, **38**:2515–2522.

Davis, S. G., Quinard, J., & Searby, 2001. A numerical investigation of stretch effects in counterflow, premixed laminar flames. *Combustion Theory and Modeling*, **5**:353–362.

Davis, S. G., Quinard, J., & Searby, 2002a. Determination of Markstein numbers in counterflow premixed flames. *Combustion and Flame*, **130**:112–122.

Davis, S. G., Quinard, J., & Searby, 2002b. Markstein numbers in counterflow, methane- and propane- air flames: A computational study. *Combustion and Flame*, **130**:123–136.

Davis, S., Law, C., & Wang, H., 1999. Propene pyrolysis and oxidation kinetics in a flow reactor and laminar flames. *Combustion and Flame*, **119**:375–399.

Day, M. S. & Bell, J. B., 2000. Numerical simulation of laminar reacting flows with complex chemistry. *Combustion Theory and Modelling*, **4**(4):535–556.

Denbigh, K., 1955. *The principles of chemical equilibrium, with applications in chemistry and chemical engineering*. Cambridge University Press.

Deshaies, B. & Cambray, P., 1990. The velocity of a premixed flame as a function of the flame stretch: An experimental study. *Combustion and Flame*, **82**:361–375.

Deshpande, M. D. & Vaishnav, R. N., 1982. Submerged laminar jet impingement on a plane. *Journal of Fluid Mechanics*, **114**:213–236.

Deville, M. O., Fischer, P. F., & Mund, E. H., 2002. *High-Order Methods for Incompressible Fluid Flow*. Cambridge University Press.

Dimotakis, P. E., 2005. Turbulent mixing. *Annual Review of Fluid Mechanics*, **37**:329–356.

Dong, Y., Vagelopoulos, C. M., Spedding, G. R., & Egolfopoulos, F. N., 2002. Measurement of laminar flame speeds through digital particle image velocimetry: Mixtures of methane and ethane with hydrogen, oxygen, nitrogen, and helium. *Twenty-Nineth Symposium (International) on Combustion*, **29**:1419–1426.

Dowd, K. & Severance, C., 1998. *High Performance Computing*. O'Reilly & Associates, Inc., second edition.

Egolfopoulos, F. N., 1994. Geometric and radiation effects on steady and unsteady strained laminar flames. *Twenty-Fifth Symposium (International) on Combustion*, pages 1375–1381.

Egolfopoulos, F. N. & Dimotakis, P. E., 2001. A comparative numerical study of premixed and non-premixed ethylene flames. *Combustion Science and Technology*, **162**: 19–35.

Egolfopoulos, F. N., Du, D. X., & Law, C. K., 1992. A comprehensive study of methanol kinetics in freely-propagating and burner-stabilized flames, flow and static reactors, and shock-tubes. *Combustion Science and Technology*, **83**:33–75.

Egolfopoulos, F. N., Zhang, H., & Zhang, Z., 1997. Wall effects on the propagation and extinction of steady, strained, laminar premixed flames. *Combustion and Flame*, **109**:237–252.

H. W. Emmons (Ed.), 1958. *Fundamentals of gas dynamics*. Princeton University Press.

Ern, A. & Giovangigli, V., 1994. *Multicomponent transport algorithms*. Springer-Verlag.

Ern, A. & Giovangigli, V., 1998. Thermal diffusion effects in hydrogen-air and methane-air flames. *Combustion Theory and Modelling*, **2**(4):349–372.

Ern, A. & Giovangigli, V., 1999. Impact of detailed multicomponent transport on planar and counterflow hydrogen/air and methane/air flames. *Combustion Science and Technology*, **149**:157–181.

Fischer, P. F., Ho, L.-W., Karniadakis, G. E., Rønquist, E. M., & Patera, A. T., 1988. Recent advances in parallel spectral element simulation of unsteady tncompressible flows. *Computers & Structures*, **30**:217–231.

Fischer, P. F. & Patera, A. T., 1991. Parallel spectral element solution of the stokes problem. *Journal of Computational Physics*, **92**:380–421.

Fischer, P. F. & Rønquist, E. M., 1994. Spectral element methods for large scale parallel navier-stokes calculations. *Computer methods in applied mechanics and engineering*, **116**:69–76.

Frenklach, M., Wang, H., & Rabinowitz, M. J., 1992. Optimization and analysis of large chemical kinetic mechanisms using the solution mapping method – combustion of methane. *Progress in Energy and Combustion Science*, **18**:47–73.

Frouzakis, C. E., Lee, J., Tomboulides, A. G., & Boulouchos, K., 1998. Two-dimensional direct numerical simulation of opposed-jet hydrogen-air diffusion flame. *Twenty-Seventh Symposium (International) on Combustion*, pages 571–577.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J., 1994. *Design Patterns*. Addison-Wesley Publishing Company.

Glarborg, P., Kristensen, P. G., Dam-Johansen, K., Alzueta, M. U., Millera, A., & Bilbao, R., 2000. Nitric oxide reduction by nonhydrocarbon fuels. implications for reburning with gasification gases. *Energy & Fuels*, **14**:828–838.

Goodwin, D. G., 2003. An open-source, extensible software suite for CVD process simulation. In M. D. Allendorf, F. Maury, & F. Teyssandier (Eds.), *Proceedings of CVD XVI and EuroCVD Fourteen,* Electrochemical Society, pages 155–162.

Gray, A., 1997. *Modern Differential Geometry of Curves and Surfaces with Mathematica*. CRC Press, second edition.

Gresho, P. M., 1991. Incompressible fluid dynamics: some fundamental formulation issues. *Annual Review of Fluid Mechanics*, **23**:413–453.

Gropp, W., Keyes, D., McInnes, L. C., & Tidriri, M. D., 2000. Globalized Newton-Krylov-Schwarz algorithms and software for parallel implicit CFD. *The International Journal of High Performance Computing Applications*, **14**:102–136.

Gropp, W., Lusk, E., & Skjellum, A., 1994. *Using MPI : portable parallel programming with the message-passing interface*. MIT Press.

Gropp, W., Lusk, E., & Skjellum, A., 1999. *Using MPI-2 : advanced features of the message-passing interface*. MIT Press.

Han, Z. & Reitz, R. D., 1995. Turbulence modeling of internal combustion engines using RNG $k - \varepsilon$ models. *Combustion Science and Technology*, **106**:267–295.

Haworth, D. C. & Jansen, K., 2000. Large-eddy simulation on unstructured deforming meshes: towards reciprocating ic engines. *Computers and Fluids*, **29**:493–524.

Henderson, R. D., 1999a. Adaptive spectral element methods for turbulence and transition, In: *High-order methods for computational physics*, T. Barth & H. Deconinck (Eds.), pages 225–324, Springer-Verlag. Lecture notes in computational science and engineering 9.

Henderson, R. D., 1999b. Dynamic refinement algorithms for spectral element methods. *Computer methods in applied mechanics and engineering*, **175**:395–411.

Henderson, R. D. & Barkley, D., 1996. Secondary instability in the wake of a circular cylinder. *Physics of Fluids*, **8**(6):1683–1685.

Henderson, R. D. & Karniadakis, G. E., 1995. Unstructured spectral element methods for simulation of turbulent flows. *Journal of Computational Physics*, **122**:191–217.

Hirasawa, T., Sung, C. J., Joshi, A., Yang, Z., Wang, H., & Law, C. K., 2002. Determination of laminar flame speeds using digital particle image velocimetry: Binary fuel blends of ethylene, n-butane, and toluene. *Twenty-Nineth Symposium (International) on Combustion*, **29**:1427–1434.

Hirschfelder, J. O., Curtiss, C. F., & Bird, R. B., 1954. *Molecular Theory of Gases and Liquids*. Wiley, New York.

Hung, P., May 2003. *Algorithms for Reaction Mechanism Reduction and Numerical Simulation of Detonations Initiated by Projectiles*. Ph.D. thesis, California Institute of Technology, Pasadena, CA.

Ianenko, N. N., 1971. *The method of fractional steps*. Springer-Verlag.

Karniadakis, G. E., Israeli, M., & Orszag, S., 1991. High-order splitting methods for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, **97**: 414–443.

Karniadakis, G. E. & Sherwin, S. J., 1999. *Spectral/hp element methods for CFD*. Oxford University Press, New York, NY.

Kee, R. J., Coltrin, M. E., & Glarborg, P., 2003. *Chemically Reacting Flow—Theory & Practice*. John Wiley & Sons, Inc., New Jersey.

Kee, R. J., Dixon-Lewis, G., Warnatz, J., Coltrin, M. E., Miller, J. A., & Moffat, H. K., 1986. Transport: A software package for the evaluation of gas-phase, multicomponent transport properties. Technical Report SAND86-8246B, Sandia National Laboratory.

Kee, R. J., Miller, J. A., Evans, G. H., & Dixon-Lewis, G., 1988. A computational model of the structure and extinction of strained, opposed flow, premixed methane-air flames. *Twenty-Second Symposium (International) on Combustion*, pages 1479–1494.

Kim, W.-W., Menon, S., & Mongia, H. C., 1999. Numerical simulations of reacting flows in a gas turbine combustor. *Combustion Science and Technology*, **143**:25–62.

Knupp, P. & Salari, K., 2003. *Verification of Computer Codes in Computational Science and Engineering*. Chapman & Hall.

Kostiuk, L. W., Bray, K. N. C., & Cheng, R. K., 1993. Experimental study of premixed turbulent combustion in opposed streams: Part I — non-reacting flow field. *Combustion and Flame*, **92**:377–395.

Kovasznay, L. I. G., 1948. Laminar flow behind a two-dimensional grid. *Proceedings of the Cambridge Philosophical Society*, **44**:58–62.

Lagerstrom, P., 1964. *Laminar flow theory*. Princeton University Press.

Lakos, J., 1996. *Large-Scale C++ Software Design*. Addison-Wesley Publishing Company.

Landau, L. D., 1944. On the theory of slow combustion. *Acta Physicochimica*, **19**:77.

Lange, T., February 19-23 2007. Will pringles fly? how P&G is using modeling & simulation to innovate innovation. Presented at SIAM Conference on Computational Science & Engineering, Costa Mesa, California. `http://www.siam.org/meetings/cse07/`.

Law, C. K., 1988. Dynamics of stretched flames. *Twenty-Second Symposium (International) on Combustion*, pages 1381–1402.

Law, C. K. & Sung, C. J., 2000. Structure, aerodynamics, and geometry of premixed flamelets. *Progress in Energy and Combustion Science*, **26**:459–505.

Leonard, A. & Wray, A., 1982. A new numerical method for the simulation of three-dimensional flow in a pipe, In *Lecture Notes in Physics, No. 170: Eighth international conference on numerical methods in fluid dynamics*, pages 335–342, Springer-Verlag.

Maas, U. & Warnatz, J., 1988. Ignition process in hydrogen-oxygen mixtures. *Combustion and Flame*, **74**:53–69.

Majda, A. & Sethian, J. A., 1985. The derivation and numerical solution of the equations for zero mach number combustion. *Combustion Science and Technology*, **42**:185–205.

Marinov, N. M., 1999. A detailed chemical kinetic model for high temperature ethanol oxidation. *International Journal of Chemical Kinetics*, **31**(3):183–220.

Markstein, G. H., 1951. Experimental and theoretical studies of flame-front stability. *Journal of the Aeronautical Sciences*, **18**:199–209.

Markstein, G. H., 1964. *Nonsteady Flame Propagation.* Pergamon Press, Oxford.

Matalon, M. & Matkowsky, B. J., 1982. Flames as gasdynamic discontinuities. *Journal of Fluid Mechanics*, **124**:239–259.

Mathur, S., Tondon, P. K., & Saxena, S. C., 1967. Thermal conductivity of binary ternary and quaternary mixtures of rare gases. *Molecular Physics*, **12**:569.

Matsushima, T. & Marcus, P. S., 1995. A spectral method for polar coordinates. *Journal of Computational Physics*, **120**:365–374.

McKee, B., June 2004. The pool gets cooler. *I.D.*, pages 78–81.

McMurtry, P. A., Jou, W.-H., Riley, J. J., & Metcalfe, R. W., 1986. Direct numerical simulations of a reacting mixing layer with chemical heat release. *AIAA Journal*, **24**(6):962–970.

Mendes-Lopes, J. M. C. & Daneshyar, H., 1985. Influence of strain fields on flame propagation. *Combustion and Flame*, **60**:29.

Miles, R. & Hamilton, K., 2006. *Learning UML 2.0*. O'Reilly & Associates, Inc.

Mohseni, K. & Colonius, T., 2000. Numerical treatment of polar coordinate singularities. *Journal of Computational Physics*, **157**:787–795.

Monchick, L. & Mason, E. A., 1961. Transport properties of polar gases. *Journal of Chemical Physics*, **35**(5):1676.

Najm, H. N. & Knio, O. M., 2005. Modeling low mach number reacting flow with detailed chemistry and transport. *Journal of Scientific Computing*, **25**:263–287.

Ortega, J. M., 1990. *Numerical Analysis A Second Course*. SIAM.

Patera, A. T., 1984. A spectral element method for fluid dynamics: Laminar flow in a channel expansion. *Journal of Computational Physics*, **54**:468–488.

Pellet, G. L., Isssac, K. M., Humphreys, W. M., Jr., Gartrell, L. R., Roberts, W. L., Dancey, C. L., & Northam, G. B., 1998. Velocity and thermal structure, and strain-induced extinction of 14 to 100% hydrogen-air counterflow diffusion flames. *Combustion and Flame*, **112**:575–592.

Pelletier, D., Turgeon, E., & Tremblay, D., 2004. Verification and validation of impinging round jet simulations using an adaptive FEM. *International journal for numerical methods in fluids*, **44**(7):737–763.

Ramshaw, J. D., O'Rourke, P. J., & Stein, L. R., 1985. Pressure-gradient scaling method for fluid-flow with nearly uniform pressure. *Journal of Computational Physics*, **58**:361–376.

Roy, C. J., 2005. Review of code and solution verification procedures for computational simulation. *Journal of Computational Physics*, **205**:131–156.

San Diego mechanism. *Chemical-Kinetic Mechanisms for Combustion Applications*, Center for Energy Research, University of California at San Diego, 2005. `http://maeweb.ucsd.edu/combustion/cermech`.

Schlichting, H., 1960. *Boundary Layer Theory*. McGraw-Hill, Inc., New York.

Shirouzu, N., 2006. Toyota review reveals need for more quality checks. *The Wall Street Journal*, (December 11):B3D.

Smith, G. P., Golden, D. M., Frenklach, M., Moriarty, N. W., Eiteneer, B., Goldenberg, M., Bowman, C. T., Hanson, R. K., Song, S., Gardiner, J., W. C., Lissianski, V. V., & Qin, Z. GRI-Mech 3.0. `http://www.me.berkeley.edu/gri_mech/`.

Smooke, M. D., Puri, I. K., & Seshadri, K., 1986. A comparison between numerical calculations and experimental measurements of the structure of a counterflow diffusion flame burning diluted methane in diluted air. *Twenty-First Symposium (International) on Combustion*, **21**:1783–1792.

Smooke, M. D., Xu, Y., Zurn, R. M., Lin, P., Frank, J. H., & Long, M. B., 1992. Computational and experimental study of oh and ch radicals in axisymmetric laminar diffusion flames. *Twenty-Fourth Symposium (International) on Combustion*, pages 813–821.

Sone, K. & Menon, S., 2003. Effect of subgrid modeling on the in-cylinder unsteady mixing process in a direct injection engine. *Journal of Engineering for Gas Turbines and Power*, **125**:435–443.

Stroustrup, B., 1991. *The C++ PROGRAMMING LANGUAGE*. Addison-Wesley Publishing Company, second edition.

Tan, Y., Dagaut, P., Cathonnet, M., Boettner, J. C., Bachman, J. S., & Carlier, P., 1994. Natural gas and blends oxidation and ignition: Experiments and modeling. *Twenty-Fifth Symposium (International) on Combustion*, **25**:1563–1569.

Tien, J. H. & Matalon, M., 1991. On the burning velocity of stretched flames. *Combustion and Flame*, **84**:238–248.

Tomboulides, A. G., Lee, J. C. Y., & Orszag, S. A., 1997. Numerical simulation of low mach number reactive flows. *Journal of Scientific Computing*, **12**(2):139–167.

Tomboulides, A. G. & Orszag, S. A., 1998. A quasi-two-dimensional benchmark problem for low mach number compressible codes. *Journal of Computational Physics*, **146**: 691–706.

Tomboulides, A. G. & Orszag, S. A., 2000. Numerical investigation of transitional and weak turbulent flow past a sphere. *Journal of Fluid Mechanics*, **416**:45–73.

Turányi, T., Zalotai, L., Dóbé, S., & Bérces, T., 2002. Effect of the uncertainty of kinetic and thermodynamic data on methane flame simulation results. *Physical Chemistry Chemical Physics*, **4**:2568–2578.

Turns, S. R., 2000. *An Introduction to Combustion*. McGraw-Hill Inc.

Vagelopoulos, C. M. & Egolfopoulos, F. N., 1998. Direct experimental determination of laminar flame speeds. *Twenty-Seventh Symposium (International) on Combustion*, pages 513–519.

Wang, H. & Frenklach, M., 1997. A detailed kinetic modeling study of aromatics formation in laminar premixed acetylene and ethylene flames. *Combustion and Flame*, **110**:173–221.

Wang, H., Laskin, A., Djurisic, Z. M., Law, C. K., Davis, S. G., & Zhu, D. L., 1999. A comprehensive mechanism of C_2H_x and C_3H_x fuel combustion. *the 1999 Fall Technical Meeting of the Eastern States Section of the Combustion Institute, Raleigh, NC*, pages 129–132.

Westbrook, C. K. & Dryer, F. L., 1981. Simplified reaction mechanisms for the oxidatioin of hydrocarbon fuels in flames. *Combustion Science and Technology*, **27**:31–43.

Williams, F. A., 1975. Recent advances in theoretical descriptions of turbulent diffusion flames, In: *Turbulent Mixing in Nonreactive and Reactive Flows*, S. N. B. Murthy (Ed.), pages 189–208.

Williams, F. A., 1985. *Combustion Theory*. Addison-Wesley Publishing Company, Redwood City, CA, second edition.

Williams, F. A., 2000. Progress in knowledge of flamelet structure and extinction. *Progress in Energy and Combustion Science*, **26**:657–682.

Wu, C. & Law, C., 1984. On the determination of laminar flame speeds from stretched flames. *Twentieth Symposium (International) on Combustion*, pages 1941–1949.

Yetter, R. A., Dryer, F. L., & Rabitz, H., 1991. A comprehensive reaction-mechanism for carbon-monoxide hydrogen oxygen kinetics. *Combustion Science and Technology*, **79**: 97–128.